STATISTICAL METHODS FOR 2D IMAGE SEGMENTATION AND 3D POSE ESTIMATION

A Thesis
Presented to
The Academic Faculty

by

Romeil S. Sandhu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2010

STATISTICAL METHODS FOR 2D IMAGE SEGMENTATION AND 3D POSE ESTIMATION

Approved by:

Professor Anthony Yezzi,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Tryphon Georgiou
Department of Electrical and Computer Engineering
University of Minnesota

Professor Allen Tannenbaum, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Linda Wills
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 15 October 2010

To my mother,

Gurjeet K. Sandhu,

You held us together, you never let us fall apart, you allowed us to pursue our dreams. This is not just my work, this is yours.

ACKNOWLEDGEMENTS

During my time here at Georgia Tech, I have been blessed with relationships with people that are impossible to be defined by simple adjectives. Your constant support and encouragement allowed me to complete my Ph.D. In particular, I would like to thank:

- **Professor Allen Tannenbaum:** On paper, you served as my thesis advisor, but in reality you did so much more. I will be forever grateful for your guidance and your friendship. Your kind and generous spirit is a model of a person that I will always strive to be. It has been the utmost honor and pleasure to have worked with you and to be your student.

- **Professor Anthony Yezzi:** I appreciate your sincerity and honesty in all aspects of life, and the time and patience you took to help guide my research.

- **Professor Tryphon Georgiou:** From day one, you never made me feel like a stranger. Aside from your valuable influence on my research, I enjoyed our many conversations we had and the laughs that we shared.

- **Professor Art Koblasz:** It was your unselfish ways that allowed me to enter into the world of research. You gave me a chance, provided encouragement, and never hesitated in helping me out.

- **Professor Jeff Shamma & Professor Linda Wills:** I appreciate you being apart of my thesis committee and the remarks that helped improve my thesis.

- **My Mentor:** Samuel Dambreville, I have enjoyed being your Padawan and will forever be indebted to the help you gave in guiding my research. In a short

time, we developed a lasting friendship, and no matter what I try to do, I know you will still be the Master Jedi. In your words, we "killed it."

- **My Friend:** Kevin Brenner, you have seen it all. It has been truly a pleasure to be your friend and am thankful for the memories we have shared over the many years at Georgia Tech.

- **My Research Crew:** James Malcolm, Shawn Lankton, and Xavier Lefaucheur. I am honored to have been able to work with you, but more importantly, I cherish the friendships we have developed. Jimi, I will always appreciate your unselfish ways and the time you took to invest with me during those long nights at coffee shops. Shawn, you taught me not to be too ridiculous and provided an anchor of sanity both in my academic and personal endeavors. Xavier, you taught me that it was ok to be ridiculous, but only during hours of conversation that took place in the morning and/or at night. Of course, Shawn, Xavier and Brenner, you guys are each the most serious, hippest, intellectual, thoughtful people I know!

- **Van Leer VIPs:** Jacqueline Trappier, Marilouise Mycko, Tasha Torrence, Christopher Malbrue, Angela Elleby. Thank y'all for your help, your collaboration, and your kindness. I hope my interests of southern food with biscuits and graaaavey did not startle you too much!

- **Members of MINERVA Lab:** Yi, Jehoon, Eli, Pete, Ivan, Behnood, Vandana, Ponnappan, Gallagher, Marc, Yogesh, Patricio, Martin, Jacob. Thank you for making the lab a friendly, scholarly, and wonderful work place.

- **My Atlanta Friends:** Christine, Eric, Rahul, David, and Nick. You guys have heard me banter about graduate school all too much, and I thank you for the support you have given to me over the years.

- **My Music:** During the long and sometimes lonely nights in the lab, you provided what coffee could not do. Thank you to the members of Pearl Jam, Semisonic, and Mason Jennings. Your creativity has helped and continues to help me spawn interesting ideas.

- **My Dad:** You instilled the value of what hard work can achieve.

- **My Brother:** Raj, I could not have asked for more. You have always been there to take care of me. You are the unsung hero in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The field of computer vision focuses on the goal of developing techniques to exploit and extract information from underlying data that may represent images or other multidimensional data. In particular, two well-studied problems in computer vision are the fundamental tasks of 2D image segmentation and 3D pose estimation from a 2D scene.

In this thesis, we first introduce two novel methodologies that attempt to independently solve 2D image segmentation and 3D pose estimation separately. Then, by leveraging the advantages of certain techniques from each problem, we couple both tasks in a variational and *non-rigid* manner through a single energy functional. Thus, the three theoretical components and contributions of this thesis are as follows:

- Firstly, a new distribution metric for 2D image segmentation is introduced. This is employed within the geometric active contour (GAC) framework.
- Secondly, a novel particle filtering approach is proposed for the problem of estimating the pose of two point sets that differ by a rigid body transformation.
- Thirdly, the two techniques of image segmentation and pose estimation are coupled in a single energy functional for a class of 3D rigid objects.

After laying the groundwork and presenting these contributions, we then turn to their applicability to real world problems such as visual tracking. In particular, we present an example where we develop a novel tracking scheme for 3-D Laser RADAR imagery. However, we should mention that the proposed contributions are solutions for general imaging problems and therefore can be applied to medical imaging problems such as extracting the prostate from MRI imagery [33].

CHAPTER I

INTRODUCTION

The field of computer vision focuses on the goal of developing techniques to exploit and extract information from underlying data that may represent images or other multidimensional data. This low-level task is then used as a basis for high-level tasks such as quality control, military surveillance, or medical image analysis. In particular, two well-studied problems in computer vision are the fundamental tasks of 2D image segmentation and 3D pose estimation from a 2D scene. It is interesting to note that while 2D-3D pose estimation and 2D image segmentation are closely related, there exist few methodologies that try to couple both tasks in a unified framework. Thus, this thesis explores not only image segmentation and pose estimation in computer vision, but takes a closer look at how one can effectively bridge both fields of interest.

1.1 Contributions and Organization of this Thesis

We begin by first introducing two novel methodologies that attempt to solve 2D image segmentation and 3D pose estimation separately, and then develop a unified framework that incorporates both tasks in a coupled manner. To further ensure the viability of the proposed algorithms in the context of image processing, we demonstrate their applicability to visual tracking.

Specifically, in Chapter 2 we present a new distribution metric for image segmentation that arises as a result in prediction theory. Forming a natural geodesic, our metric quantifies "distance" for two density functionals as the standard deviation of the difference between logarithms of those distributions. Using level set methods, we show the energy based on the metric can be incorporated into the geometric active contour (GAC) framework.

Then, in Chapter 3, we propose a particle filtering approach for the problem of registering or estimating the pose of two point sets that differ by a rigid body transformation. Due to the fact that registration algorithms usually compute the transformation parameters by maximizing a metric given an estimate of the correspondence between points across the two sets of interest, we approach the task as a posterior estimation problem. From this, the corresponding distribution can naturally be estimated using a particle filter.

Consequently, both Chapter 2 and 3 introduce the fundamentals for Chapter 4, whereby leveraging the advantages of certain techniques from each problem, we couples both tasks in a variational and *non-rigid* manner through a single energy functional. In general, most frameworks that couple both pose estimation and segmentation assume that one has the exact knowledge of the 3D object. In other words, this assumption may be violated in non-ideal conditions if only a general class to which a given shape belongs is given (e.g., cars, boats, or planes). The proposed methodology also accomplishes the non-rigid task of pose estimation and segmentation without the need for point correspondences or specific constraints on the type of 3D shape.

After laying the theoretical groundwork in Chapter 2, 3, and 4, we demonstrate the applicability of the proposed algorithms in real-life applications in Chapter 5. That is, we present visual tracking algorithms capable of providing aim-point maintenance for 3-D Laser RADAR (3DLADAR) imagery as well as for autonomous mortar tracking. More importantly, given that the algorithms presented in this thesis are for general image processing, we also mention several medical imaging problems (e.g., extracting the prostate from magnetic resonance (MR) imagery) in which one has or can utilize the proposed algorithms to accomplish the task at hand.

Thus, the contributions of this thesis will be organized as follows:

- **Chapter 2:** Presents a new distribution metric for image segmentation that

arises as a result in prediction theory and incorporate this into the GAC framework. Experimental segmentation examples are provided and results are given.

- **Chapter 3:** Presents a particle filtering approach for point set registration and pose estimation in which we exploit the underlying dynamics or uncertainty associated in any registration problem. Experimental results are given to highlight algorithm's robustness to challenging registration problems.

- **Chapter 4:** Presents a unified and variational approach in which we couple both 2D image segmentation and 3D pose estimation in a single energy functional for a class of 3D rigid objects. Experimental results demonstrate the effectiveness of the algorithm with regards to either 2D image segmentaion or 3D pose estimation.

- **Chapter 5:** Presents a real-world problem of tactical tracking using 3DLADAR imagery for which the proposed algorithms provide a robust solution.

- **Chapter 6:** Offers a summary of the presented work, draws conclusions from the thesis, and discuses possible directions for future work.

However, before we present the main contributions of this thesis, we must first revisit some of the key results that have been made pertaining to the above fields of interest.

1.2 Literature Review for Image Segmentation

The general segmentation problem, which falls under the category of geometric source separation, involves separating data into N distinct partitions via decision boundaries. However, a piecewise assumption of two sets is generally made for the particular case of images. Using this assumption, image segmentation can be defined as optimally partitioning a scene into an "object" and a "background" [8]. In particular, we will restrict our approach of segmentation to that of the geometric active contour framework, whereby a curve is evolved continuously until it satisfies a stopping criterion

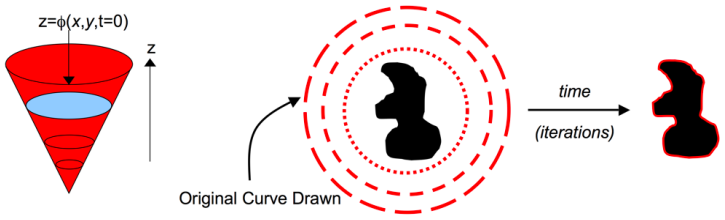


Figure 1: Illustration of level set methods. On the left is the implicit curve representation. On the right, the general scheme of how an active contour deforms to segment an object over time.

that coincides with the object's boundaries as shown in Figure 1 [51, 52, 13, 8, 104, 63].

In employing this methodology, a curve is represented as the zero level-set of a higher dimensional surface [88, 64]. A common choice is the signed distance function. Although this implicit representation of a curve is computationally more expensive then parametric approaches, it allows for the contour to naturally undergo topological changes. Specifically, certain variational approaches rely on characterizing the object by local features such as edges to drive the curve evolution; see [13, 52] and the references therein. However, these edge-based techniques were shown to be susceptible to noise and missing information. Consequently, an alternative characterization, based on so-called "region-based methods," is to assume the "object" and "background" possess differing image statistics (see [14, 66, 72]).

In this thesis, we will focus on region based approaches due to their high level of robustness to noise and initialization when compared to models based on local information. If we restrict our discussion to approaches that generalize the statistical inference beyond first and second moments to entire probability density functions (pdf), segmentation can be reinterpreted as measuring the "distance" between two distributions via a similarity metric.

That is, by directly measuring the discrepancies of pixel intensities, Rousson and Paragios proposed to maximize the L_2-distance between the log-likelihood of two distributions defined by the interior and exterior regions of the segmenting curve [66].

Although a Gaussian assumption is made in their work, an extension to entire probability distributions is straightforward. Interestingly, the metric proposed in Chapter 2, when mapped to a linear space, results in a similar energy [36]. Recently, Freedman *et. al* introduce the Bhattacharrya distance and Kullback-Leibler divergence for segmentation by maximizing the similarity between the density of a region enclosed by a curve $\mathcal{C}$ and a known pdf that is learned *a-priori* [32]. By relaxing the assumption of *a-priori* knowledge, Rathi *et. al* derived a flow that optimally separates distributions via the Bhattacharrya measure [72]. Although these measures are based on varying disciplines and are motivated by a specific problem, they can be extended to a general class of densities. To this end, Chapter 2 introduces a new distribution metric for image segmentation that first arose in prediction theory, but will be shown to be equally effective in image segmentation.

Lastly, we should note that in formulating an image segmentation problem via a region-based method, an assumption of separable statistics is made. That is, one generally assumes the object exhibits a homogenous profile with regards to the background, and the object of interest can be indirectly identified with only image information. However, this assumption may not hold due to clutter or occlusions. This has resulted in the proposed use of a shape prior to restrict the evolution of the active contour [55, 21, 15, 23, 94]. While not as relevant in Chapter 2, the notion of a shape prior will be essential in formulating our unified framework as presented in Chapter 4. Also, an understanding of pose estimation will be essential. Necessary background information regarding pose estimation is given next.

1.3 Literature Review for Point Set Registration and Pose Estimation

Pose Estimation is concerned with relating the spatial coordinates of an object in the 3D world (with respect to a calibrated camera) to that of a 2D scene or another 3D object. In this proposal, we have subdivided the problem into two areas: 3D (or 2D)

(a) (b) (c)

Figure 2: Common problems in point set registration. (a) Initial alignment that can yield an incorrect registration to the "wrong" side of the truck, when using iterative based techniques. (b) Dense point set. (c) Sparse point set.

point set registration and 2D-3D pose tracking. We begin with the classical problem of registering or estimating the pose of two point sets.

By decoupling the problem of estimating correspondences between two point sets and their transformation parameters, Besel and McKay [5] introduce the well-known Iterative Closest Point (ICP) algorithm. Given an initial alignment, ICP assigns a set of correspondences based on the L_2 distance, computes the transformation parameters, and then proceeds in an iterative manner with a newly updated set of correspondences. However, the basic approach is widely known to be susceptible to local minima. To address this issue, Fitzgibbon [31] introduces a robust variant by optimizing the cost function with the Levenberg-Marquardt algorithm. Even though this method and variants of ICP [17, 31] do improve the narrow band of convergence, they are still heavily dependent on the initial alignment, and may fail due to the existence of homologies (due to noise, clutter, outliers) within the correspondence matrix. For instance, Figure 2 demonstrates a common problem in registration, in which a poor initial alignment can yield an incorrect registration to the "wrong" side of the truck.

To overcome the problem of sensitivity to initialization, a second class of point set registration schemes, referred to as "shape descriptors," has emerged in the graphics

community [34, 49, 46]. Typically, these approaches introduce structural information into the registration scheme. This allows them to perform well under poor initializations as well as handle partial structures or missing information. Unfortunately, these techniques are generally ill-suited for tasks such as tactical tracking, whereby the point set density is unknown and can be adjusted during the acquisition phase. Without special consideration, registration may fail if one tries to match a sparse cloud to a dense cloud. See Figure 2 for an illustration of "sparsity."

Another proposed methodology for solving the problem of dependency on initial alignment (as seen standard in ICP and other iterative based methods) is the Robust Point Matching (RPM) algorithm [19, 18]. This approach performs an exhaustive search that is reduced over time with an appropriate annealing schedule. However, the authors of [91] demonstrate the failure of RPM in the presence of clutter or when certain structures are missing.

The use of robust statistics and measures form the next class of point set registration algorithms [47, 92]. Specifically, representing point sets as probability densities, Tsin and Kanade [95] propose a Kernel Correlation (KC) approach using kernel density estimates. The method computes the optimal alignment by reducing the "distance" between sets via a similarity metric. An extension is considered in [48] through the use of a Gaussian mixture model. In particular, both of these approaches propose a registration technique without explicitly establishing point correspondences between data sets, and both methods can be considered as multiply-linked ICP registration schemes. While this allows for a wider basin of convergence than traditional ICP-like algorithms, one can see that the approaches become computationally expensive as one point set must interact with each point in the opposing set. Moreover and more importantly, to overcome poor initializations or missing information, the KC algorithm must use a kernel with a larger bandwidth. This effectively "smoothes" the data sets, and results in an alignment of distributions spatially. Hence, there

is a trade-off between the point-wise accuracy (increases with smaller bandwidth) and its dependency on initial alignment or missing information (decreases with larger bandwidth).

Consequently, a natural extension would be to employ a multi-scale approach as proposed by Granger and Pennec [40]. Their algorithm begins by aligning the center of mass of each point set, and then proceeds with the lowering of a "smoothing" factor to ensure the point-wise accurate feature of ICP. We should note that the framework presented in this Chapter 3 shares certain similarities with this method, notably the general notion of having a global-to-local approach as well as invoking a robust objective functional through the use of Gaussian mixture model. However, while [95, 48] can be re-interpreted as a multiply linked ICP registration scheme, our proposed algorithm can be considered as a switching stochastic ICP approach where one point set interacts only with a handful of correspondences. Thus, we keep the explicit establishment of point correspondences as in the iterative techniques, which ensures, on the local level, an accurate ICP-like algorithm. Moreover, we do not require an annealing schedule in our global-to-local approach such as that proposed in [40], since this is naturally embedded in the diffusion process of the prediction model. In Chapter 3, we compare the KC algorithm with the particle filtering technique discussed in this present work.

Related results that follow the approach presented in Chapter 3 are based on filtering methods [56, 62]. Ma and Ellis [56] pioneered the use of the Unscented Particle Filter (UPF) for point set registration. Although the algorithm accurately registers small data sets, it requires a large number of particles (5000) to perform accurate registration. Because of the large computational costs involved using large sample sizes, the method becomes impractical for large data sets. To address this issue, the authors in [62] propose to use an Unscented Kalman Filter (UKF) approach. However, their method suffers the limitation of assuming a unimodal probability

distribution of the state vector, and thus, may fail for multimodal distributions.

In contrast to point set registration, 2D-3D pose tracking involves estimating the 3D pose from a single or multiple 2D images [42, 57]. Although a complete literature review is beyond the scope of this proposal, most methodologies can be described as follows. First, one chooses a local geometric descriptor (e.g., points [70], lines [28, 59], or curves [30, 78]) or image intensity [9] that can best quantify features on the image to its corresponding 3D counterpart. Then, explicit point correspondences are established in order to solve for the pose transformation. As with most correspondence-based algorithms, which rely on local features, it can be readily seen that these techniques may suffer from the existence of homologies like that of point set registration.

Aside from the non-trivial task of establishing correspondences, many 2D-3D pose estimation techniques make certain (sometimes rather restrictive) assumptions on the class of shapes that they can handle. Recently, the authors in [78] propose to relax such restrictions by focusing on free-form objects. However, even this type of algebraic approach may become increasingly difficult to estimate the pose for an arbitrary or complex shape. Moreover, and more importantly, the above methods typically constrain their approaches to the knowledge of a pre-specified 3D model. To overcome this constraint, non-rigid algorithms have appeared in the area of human pose estimation [26, 77, 4]. While we should note that the focus of pose estimation in this thesis is not specific to this area of computer vision, the framework developed in Chapter 4 is closely related if one were to learn a large class of deformations as opposed to rigid objects. However, in contrast to the methods such as [26, 77], the algorithm discussed in Chapter 4 relies on the surface differential geometry of a 3D model. This allows us to eliminate the need for point correspondences altogether while still being able to deal with a complex shape. It is at this point that we mention that one of the key goals of this thesis is to combine the strengths (and circumvent the

weaknesses) of pose estimation and image segmentation for a wide class of 3D objects with few restrictions as possible. Before doing so, we first review existing algorithms that attempt to solve this important problem in a unified manner.

1.4 Literature Review Joint 2D-3D Pose Estimation and 2D Image Segmentation

It is interesting to note that while 2D-3D pose estimation and 2D image segmentation are closely related, there exist few methodologies that try to couple both tasks in a unified framework. An early attempt to solve the problem of viewpoint dependency for differing aspects of a 3D object is given in [74]. In their work, the authors propose a region-based active contour that employs a unique shape prior, which is represented through a generalized cone based on single reference view of an object. Although the method performs well under different changes in aspect, it is not able to cope with a view of an object that is substantially different from the reference view.

In addition, even though we have restricted our discussion of this proposal to the GAC framework, we note that recent work has been done in simultaneous pose estimation and segmentation via dynamic graph cuts [11, 53]. In their approach, the authors propose an articulated shape prior through a stick-man or skeleton model. Then, in order to capture deformations occurring to the object, one must optimize over a set of pre-defined parameters corresponding to specific motions in a model's movement. In relation to this work, our focus would be to accurately quantify the deformation through a set of 3D models like that of [4] through statistical learning techniques. More importantly, we proposed an algorithm that incorporates a general class of shapes for which one may not be able to associate a skeleton model. This is shown in Figure 3. Also, the above methodologies differ in the segmentation approach used (i.e. graph cuts versus active contours), in which we note that each method has its advantages and disadvantages.

While it is not the intended contribution of Chapter 4, one could also alternatively

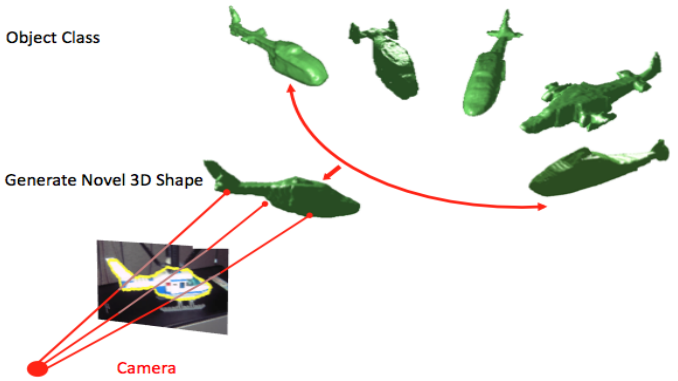


Figure 3: Chapter 4's non-rigid approach to 2D image segmentation and 3D pose estimation through the use of multiple 3D shapes.

view the proposed framework as 3D reconstruction from a single or multiple 2D images. Although this area is abundant with methodologies, we refer the reader to several works that propose to solve this difficult task [10, 90, 41, 79]. In particular and like that of the algorithm presented in Chapter 4, [90] proposes to solve this task by inducing a prior on the possible reconstructions. However, aside from using several views, the prior does not incorporate information about a specific class of objects, but rather is employed as a prior for smoothness. Recent work by [79] utilize a probabilistic graph model to reconstruct an object of a certain class from a single 2D view. A key difference between the method in Chapter 4 and that of [79] is the manner in which we approach the task itself. That is, although we use the image as measure of fidelity, we do not incorporate a graphical model. Nevertheless, we do not consider this to be a limitation, but rather a philosophical difference.

In relation to the proposed unified framework for pose estimation and segmentation, the authors in [76, 86] also propose a solution to solve the joint task of pose estimation and segmentation for the case of *rigid* objects. In [76], the authors account for a variation in the projection of the 3D shape by evolving an active contour in conjunction with the 3D pose parameters to minimize a joint energy functional. While this is less restrictive, the algorithm optimizes over an infinite dimensional active contour as well as the set of finite pose parameters. Moreover, in order for

one to determine the shape prior and the corresponding 3D pose, costly back projections must be made through ICP-like correspondences. An extension is considered in [86], whereby the authors successfully eliminate the need to evolve the active contour by performing a minimization of 3D pose parameters instead. However, the costly back-projections and correspondences remain.

In Chapter 4, we derive a variational framework to jointly segment a *rigid* object in a 2D image and estimate the corresponding 3D pose through the use of a 3D shape prior. We then show that the method can be readily extended to a 3D class of rigid objects, in which the objects themselves are (non)linearly related. Specifically, our algorithm uses a region-based segmentation method to continuously drive the pose estimation process. Using region-based segmentation results in a global approach, which avoids using local features or ICP-like correspondences by relying on surface differential geometry to link geometric properties of the model surface and its corresponding projection in the 2D image domain. The methodology is motivated by similar approaches that were originally constructed for stereo reconstruction from multiple cameras [101, 102] and further extended for camera calibration [96].

1.5 Literature Research for Visual Tracking

As stated previously, while Chapter's 2, 3, and 4 lay the theoretical groundwork for the general proposed computer vision algorithms, Chapter 5 discusses the applicability of those algorithms with respect to visual tracking. For the sake of the completeness, we provide a survey of existing methods in addition to those already discussed that focus on solving the visual tracking problem.

Visual tracking has been a significant topic of research in the field of computer vision; see [8, 45, 89, 103] and references therein. The ultimate goal of visual tracking is to continuously identify the 3D location of an object of interest from an image

sequence. However, due to the difficulty of developing a tractable solution for estimating the 3D position from a 2D scene, many researchers have tacitly restricted the tracking problem to be concerned with only the relative 2D location of the object in which segmentation is often employed in conjunction with Kalman or particle filters [50, 84].

We consider those methods that employ various filtering schemes such as the Kalman filter [93], unscented Kalman filter [50, 97], and particle filter [29, 84]. Specifically, the authors in [69, 16] employ a finite dimensional parameterization of curves, namely B-slines, in conjunction with the unscented Kalman filter for rigid object tracking. Generalizing the Kalman filter approach, the work in [99] presents an object tracking algorithm based on particle filtering with quasi-random sampling. Since these approaches only track the finite dimensional group parameters, they cannot handle local deformations of the object.

As a result, several tracking schemes have been developed to account for deformation of the object via the level set technique. In relation to this thesis, some early attempts for 2D visual tracking using level set methods can be found in [65, 100]. In particular, authors in [100] propose a definition of motion for a deformable object. This is done by decoupling an object's motion into a finite group motion known as "deformotion" with that of deformation, which is any departure from rigidity. Building on this, authors in [73] introduce a deformable tracking algorithm that utilizes the particle filtering framework in conjunction with geometric active contours. Other approaches closely related to these frameworks are given in [93, 67, 68]. Here the authors use a Kalman filter for predicting possible movements of the object, while the active contours are employed only for tracking deformations of the corresponding object.

Overall Contribution: One of the key questions that is to be ultimately addressed in this thesis *is how can one fully exploit the knowledge of a single 3D model*

or a general class of 3D shapes for the purpose of 2D image segmentation and/or 3D pose estimation?

CHAPTER II

A NEW DISTRIBUTION METRIC FOR 2D IMAGE SEGMENTATION

In this chapter, we present a new distribution metric for image segmentation that arises as a result in prediction theory. This chapter is based on [83] and is organized as follows: In the next section, we provide a brief overview of the metric proposed for image segmentation. In Section 2.2, we describe how one can cast the metric in the geometric active contour framework. Experimental results are given in Section 2.3. Lastly, we conclude with Section 2.4.

2.1 Similarity Metric via Prediction Theory

Let us first sketch and review some brief concepts that are associated with metric for image segmentation. We seek to maximize the distance of two distributions defined by a certain photometric variable. In other words, in segmentation we look to "predict" the distribution that best characterizes the object or foreground. Originally motivated by measuring the similarity between spectral distributions, the metric proposed by [36] is derived based on principals in prediction theory. That is, let $f_1(\theta)$ and $f_2(\theta)$ denote the power spectral distribution defined on the interval $\theta \in [0, \pi]$. These distributions correspond to their respective zero-mean random stationary processes $u_k^{f_i}$ with $i \in \{1, 2\}$ and $k \in \mathbb{Z}$. Then the variance of the linear or one-step-ahead prediction error for a given process of spectral distribution $f_i(\theta)$ is

$$\mathcal{E}\{|u_0^{f_i} - \hat{u}_{0|past}^{f_i}|^2\} = \mathcal{E}\{|u_0^{f_i} - \sum \alpha_k^{f_i} u_{-k}^{f_i}|^2\} \tag{1}$$

with $k > 0$ and where $\alpha_k^{f_i}$ are the coefficients that minimize the linear prediction error variance for the specific $f_i(\theta)$. Now suppose that we are given a known power

distribution $f_1(\theta)$, and we would want to measure the "distance" or similarity with another spectral density $f_2(\theta)$. This can be achieved by first assuming the density $f_2(\theta)$ originates from the distribution $f_1(\theta)$. From this, we use $f_2(\theta)$ to design a predictor and compare how well it performs **against the optimal prediction** that is based on $f_1(\theta)$. Hence, we arrive at the following measure denoted as the *degradation of predictive error variance*

$$\rho(f_1, f_2) = \frac{\mathcal{E}\{|u_0^{f_1} - \sum_{k=1}^{k=\infty} \alpha_k^{f_2} u_{-k}^{f_1}|^2\}}{\mathcal{E}\{|u_0^{f_1} - \sum_{k=1}^{k=\infty} \alpha_k^{f_i} u_{-k}^{f_1}|^2\}}. \tag{2}$$

Equation (2) gives us a basis for measuring (dis)similarity and can be viewed as analogous to "divergences" such as the Kullback-Leibler entropy seen in Information Theory [20]. Considering infinitesimal perturbations between a spectral density f and $f + \Delta$ on finite set χ, we arrive at the following pseudo-metric defined on the cone of power spectral density functions.

$$g_f(\Delta) := \int_\chi \left(\frac{\Delta(x)}{f(x)}\right)^2 dx - \left(\int_\chi \frac{\Delta(x)}{f(x)} dx\right)^2. \tag{3}$$

Given that g_f is insensitive to scaling, we note that it is consider a psuedo-metric. However, we will refer to it as a metric for the sake of clarity. Moreover, equation (9) is the corresponding geodesic distance given in the level set framework. While the derivation of the metric along with the computation of the minimal geodesic path is beyond the scope of this thesis, we do refer the interested reader for an enlightened discussion on its original development [35, 36]. Next, we review popular distribution functionals and compare the Riemannian structure induced by both the Fisher metric and the metric just proposed.

2.1.1 Fisher Metric, Hellinger Discrimination, Bhattacharrya Distance

Motivated by the prediction problem for spectral densities, the similarity metric proposed in the previous section originated from the measure denoted as the degradation of predictive error variance. In a similar fashion, the notion of degradation has been

applied to coding efficiency [20]. In other words, let us first focus on the differential geometry of the finite dimensional simplex $\mathcal{P} := p(k)$ where $k \in \{1, ..., n\}$ such that $p(k) > 0$ and $\sum_x p(x) = 1$. Figure 4b displays the simplex as a red triangular surface for $k = 3$. From this, the optimal code length for a random source of independent symbols generated according to $p_1 \in \mathcal{P}$ is given by $-\sum_x p_1(x) \log(p_1(k))$. However, if the code is based on the wrong choice between two alternatives $p_1, p_2 \in \mathcal{P}$, i.e., when the code is designed **based** on p_2 while the symbols are generated according to p_1, we arrive at the following measure denoted as the degradation of coding efficiency

$$K(p_1, p_2) := \sum_k p_1(k) \log(\frac{p_1}{p_2}). \tag{4}$$

It is important to note here that this measure of degradation is analogous to equation (2), and equation (4) is better known as the Kullback-Leibler (KL) divergence. Now if we take infinitesimal perturbations as before, the KL divergence induces a Riemannian structure known as the Fisher information metric

$$g_{fisher,p}(\Delta) = \sum_\chi \frac{\Delta(k)^2}{p(k)}. \tag{5}$$

Let us now consider the mapping $p(k) \mapsto u(x) := 2\sqrt{p(x)}$, which maps the probability simplex onto the sphere $\sum_\chi u(x)^2 = 4$. Figure 4 demonstrates this mapping as the simplex (red triangular surface) is mapped to the blue spherical surface. Interestingly, this map turns out to correspond to distances measured by the Fisher metric as well as Euclidean distances [3, 12]. That is, if the end points are chosen, the natural geodesic distance is known to be the Hellinger discrimination [43]

$$H(p_1, p_2) = \sum_\chi \left(\sqrt{p_1(k)} - \sqrt{p_2(k)}\right)^2. \tag{6}$$

Moreover, because geodesics are mapped to great circles, and the distances measured by the Fisher metric on the probability simplex corresponds to the length of arcs on the sphere, we can arrive at the famed Bhattacharrya coefficient [7]

$$B(p_1, p_2) = \sum_\chi \sqrt{p_1(k)p_2(k)}. \tag{7}$$

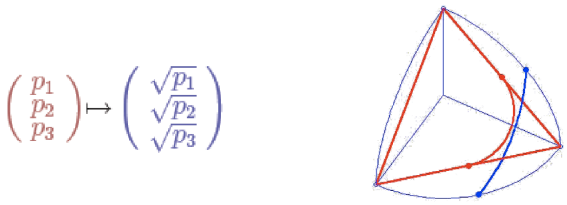


Figure 4: (a) The map $p \mapsto \sqrt{p}$ for each point on the simplex (b) The simplex as a triangular red surface taken onto the orthant of the blue spherical surface

This distance functional is the cosine of the geodesic arc between the two image points under the mapping $p \mapsto \sqrt{p}$. In addition, the above distances can be extended to continuous functions, which results in a integration over the density rather than a summation. In the next section we draw upon parallels and differences of both metrics.

2.1.2 Parallels and Comparison to Information Geometry

Given that we seek to measure the distance between two density functionals, the notion of degradation of performance is a powerful tool in forming a measure of similarity. Moreover, both degradation measures induce a Riemannian metric from their respective probability distributions. However, the manner in which the respective metrics penalize perturbations and distances vastly differs as seen Table 1. Also, while the $p \mapsto \sqrt{p}$ maps the probability simplex onto the sphere, the mapping $f \mapsto \log f$ takes the cone of spectral densities into a linear space. In other words, we are able to map geodesics defined by our Riemannian metric into straight lines. In doing so, one should realize the more popular energy functional proposed by Rousson and Paragios [66] can be now reformulated and is similar to that of the mapped version of our Riemannian metric (in the linear L_2 sense). Table 1 below explains the several differences between the Fisher metric and the metric proposed in this chapter. In the next section, we cast the geodesic distance as the energy functional in the GAC framework for image segmentation.

Type of Comparison	Information-Based Metric	Prediction-Based Metric
metric	$\int \frac{\Delta^2}{p}$	$\int (\frac{\Delta}{f})^2 - (\int \frac{\Delta}{f}))^2$
mapping	$p \mapsto \sqrt{p}$	$f \mapsto \log f$
geodesic distance	great circles	logarithmic families

Table 1: Theoretical Comparison between the Fisher Information Metric and our proposed Metric

2.2 Proposed Framework

We consider the problem of segmenting an image I. That is, we first assume the image is composed of two homogeneous regions referred to as "object" and "background". From this, the goal of segmentation is to capture these two regions. To do so, we enclose a curve $\mathcal{C}$, which is represented as the zero-level set of a signed distance function $\phi : \Re^2 \rightarrow \Re$ such that $\phi < 0$ represents the inside of $\mathcal{C}$ and $\phi > 0$ represents the outside of $\mathcal{C}$. Our goal is to evolve the curve $\mathcal{C}$, or equivalently ϕ, so that the interior matches the object and the exterior matches the background. The curve $\mathcal{C}$ would then match the boundary $\partial\Omega$ separating the object and background. The general minimization is performed by evolving C according to the flow:

$$\frac{\partial \phi}{\partial t} = \nabla_\phi E_{image} + \lambda \cdot \delta(\phi) \cdot \text{div}\left(\frac{\nabla(\phi)}{\|\nabla(\phi)\|}\right), \tag{8}$$

where the second term is incorporated such that the curve remains "smooth." We now propose an energy functional based on the metric discussed previously [35] and derive the corresponding partial differential equation (PDE) that describes its curve evolution in the level set framework. Moreover, because the metric measures similarity or "distance" as the standard deviation between the log-likelihood of two distributions, p_{in} and p_{out}, we seek to maximize the following energy functional

$$E_{image}(z, \phi) = \sqrt{\int_z \left(z \log \frac{p_{in}(z,\phi)}{p_{out}(z,\phi)}\right)^2 dz - \left(\int_z z \log \frac{p_{in}(z,\phi)}{p_{out}(z,\phi)} dz\right)^2} \tag{9}$$

where $z \in \mathcal{Z}$ is the photometric variable, and p_{in} and p_{out} are the pdf's defined on the random variable z. In the present work, we restrict the variable z to set of gray

level values $\{1, 2, ..., 256\}$. Moreover, let $I : \Re^2 \rightarrow \mathcal{Z}$ be a mapping of the image defined over the domain Ω to the photometric variable z, and let $x \in \Re^2$ be the image coordinates. Then the pdf inside and outside the curve $\mathcal{C}$ can be formulated as

$$p_{in}(z,\phi) = \int_{\Omega} \frac{K(z - I(x))H(-\phi)}{H(-\phi)} dx \quad \text{and} \quad p_{out}(z,\phi) = \int_{\Omega} \frac{K(z - I(x))H(\phi)}{H(\phi)} dx$$

where $K(z - I(x))$ is a specified kernel. For numerical experiments, we have used $K(z - I(x)) = \delta(z - I(x))$. Also $H_\epsilon : \mathbf{R} \mapsto \{0, 1\}$ denotes the Heaviside step function with the corresponding derivative δ_ϵ. These are both given as follows

$$H_\varepsilon(\phi) = \begin{cases} 1 & \phi < \epsilon \\ 0 & \phi > \epsilon \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\epsilon})) & \text{otherwise} \end{cases} \qquad \delta_\epsilon(\phi) = \begin{cases} 0 & \phi < \epsilon, \quad \phi > \epsilon \\ \frac{1}{2\epsilon}(1 + \cos(\frac{\pi\phi}{\epsilon})) & \text{otherwise} \end{cases}$$

The gradient $\nabla_\phi T$ can be computed using the calculus of variations. Taking the first variation with respect to ϕ and noting the expected value of the variable z is defined as $\mathcal{E}\{f(z)\} = \int_z z \cdot f(z)$, we arrive at the following PDE

$$\nabla_\phi E_{image} = -\frac{\delta_\epsilon(\phi)}{T} \cdot [\mathcal{E}\{B \cdot G\} - \mathcal{E}\{B\} \cdot \mathcal{E}\{G\}]. \tag{10}$$

with B and G given as

$$B = \log \frac{p_{in}(z,\phi)}{p_{object}(z,\phi)} \quad G = \left[\left(\frac{1}{A_{in}} + \frac{1}{A_{out}} \right) - K(z - I(x)) \left(\frac{1}{A_{in} p_{in}(z,\phi)} + \frac{1}{A_{out} p_{out}(z,\phi)} \right) \right]$$

Also, if A_{in} is given by $\int_\Omega H_\epsilon(\phi) dx$, then equation (10) is a PDE that describes the evolution of the curve $\mathcal{C}$ that optimally maximizes the "distance" between the distribution exterior to the segmenting curve with that of the pdf inside the curve.

2.3 Experiments

In this section, we present experimental results obtained from evolving a curve $\mathcal{C}$ according to Equation (8). Moreover, we provide a qualitative comparison between our metric and that of the result obtained with the Bhattacharrya measure. The

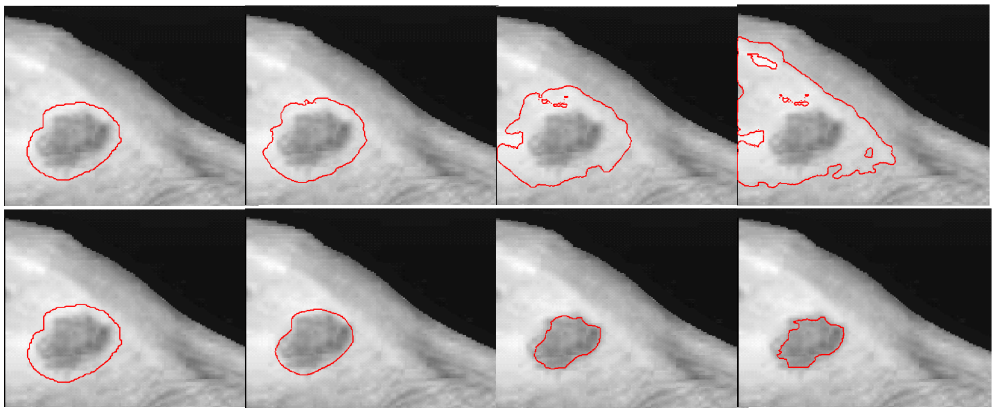


Figure 5: Case one of the Kaposi Sarcoma. Top Row: shows the evolution of a curve according to the Bhattacharrya distance resulting in an unsuccessful segmentation. Bottom Row: Successful segmentation when using the proposed similarity metric

segmenting comparisons are done on images representing patients whose skin are infected with Kaposi Sarcoma. We also demonstrate the proposed algorithm on several other classical images including the Corpus Collusom.

2.3.1 Comparative Segmentation: Kaposi Sarcoma

Although various metrics and distributional functionals have been proposed for segmentation in the GAC framework, a qualitative comparison to demonstrate the varying segmentation behavior has not been done (to the best of our knowledge) with regards to similarity measures. The goal of this experiment is not to claim the ideal energy model for distinguishing between two distributions, but to add our energy to a general class of models discriminating on probability densities. Hence, we would like show that with the same distribution, two different metrics can provide different results. Note, no smoothing or regularization term is used in these comparisons, and the results are obtained strictly on the energy describing the respective similarity measure.

In Figure 5, we demonstrate a segmentation comparison between the flow derived from the Bhattacharrya measure and the similarity metric proposed in this chapter.

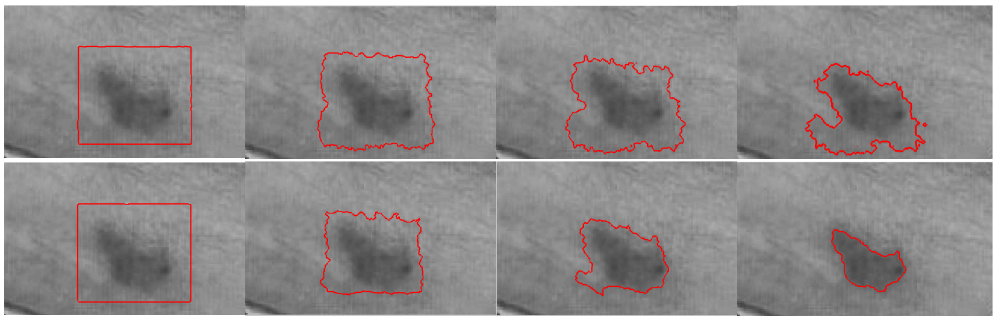


Figure 6: Case two of the Kaposi Sarcoma. Top Row: shows the evolution of a curve according to the Bhattacharrya distance resulting in an unsuccessful segmentation. Bottom Row: Successful segmentation when using the proposed similarity metric

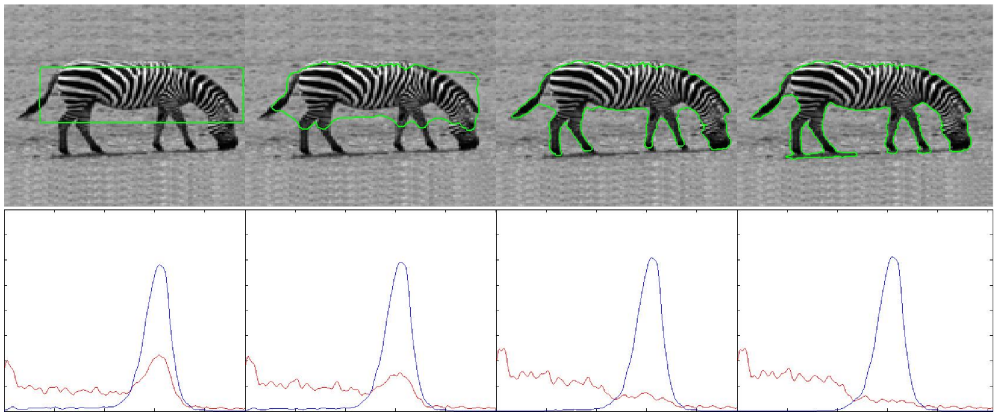


Figure 7: The Classic Zebra. Top Row: Several stages of the segmentation in which we capture the bimodal object Bottom Row: Corresponding distribution plots of both interior (red) and exterior (blue) regions if segmenting curve

Note, the same initialization is used. The segmentation result obtained by discriminating distributions with the Bhattacharrya measure fails to capture the infected portion of the skin. Moreover, it favors to segment an entirely different region. However, an acceptable segmentation result is obtained by evolving the curve according to equation (10). Initial, intermediate, and final segmentation results are shown.

On a different case of the Kaposi Sarcoma, Figure 6 shows that the Bhattacharrya distance is again unable to capture the infected portion of the skin while the proposed energy results in a successful segmentation. From these experiments, we believe (without proof) the ability to capture objects under low contrast is a major qualitative

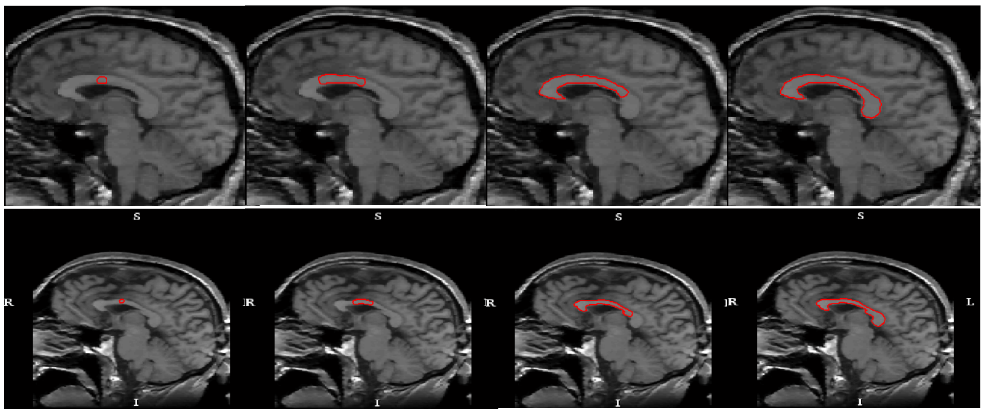


Figure 8: Corpus Callosum. Top Row: Successful Segmentation of a Corpus Callosum, a generally challenging task without discriminating on the entire pdf . Bottom Row: Successful segmentation of another case of the Corpus Callosum.

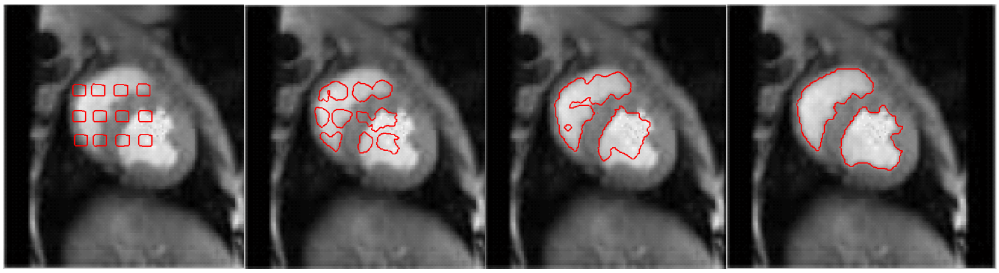


Figure 9: Successful segmentation of an MRI image of a heart using a different initialization.

difference between the energy proposed and the Bhattacharrya measure. As present in both of the Kaposi Sarcoma images, the infected portion's gray-scale intensity is not entirely different from the neighboring region. Several stages of the segmentation are given.

2.3.2 Segmentation Results: Medical Structures, Classic Zebra

In this section, we test the our region based segmentation model on several images, which further demonstrates the viability of using the proposed energy for image segmentation. A common example, which is often tested with energy models that discriminate on probability distributions, is the zebra image. The goal here is to capture the entire zebra by separating the distributions such that we obtain a bimodal object

with a unimodal background. We note that several segmentation methods have been able to capture this image. However, for the sake of completeness, we show results in Figure 7. Stages of the segmentation are shown along with the corresponding plots of the probability distributions.

Moreover, segmenting biological structures from medical images is often a challenging task. This is due to the inherent inhomogeneous distribution of a photometric variable as well as the low contrast and noise (as seen in the Kaposi Sarcoma). In the remaining examples, we segment both the corpus callosum and an MRI image of a heart. With different types of initialization, we are able to capture the finer details as shown in Figure 8. It should be noted that without discriminating on the entire probability distribution (e.g., making a Gaussian assumption), one would not be able to segment the corpus callosum. Finally, in Figure 9, we demonstrate our algorithm by segmenting an MRI image of a heart with a different type initialization compared to other initializations seen in this chapter.

2.4 Chapter Conclusion

In this chapter, we introduced a new metric for image segmentation that quantifies the "distance" between two distributions as the standard deviation of the difference between logarithms of those densities. Although several metrics and measures have been introduced for image segmentation, results may vary. The resulting behavior can be attributed to the fact that the respective metrics penalize perturbations on a manifold of probability densities in a different manner.

Moreover, although the results presented in this chapter yield acceptable segmentation results, the proposed algorithm does have inherent drawbacks. As mentioned in Chapter 1, if the basic assumption of separable statistics does not hold due to occlusions or clutter surrounding the object, the active contour will over(under)-segment the object of interest. This can be mainly attributed to the fact that one only evolves

a curve according to image information alone. Thus, the incorporation of prior information in the form of a shape prior is usually used. However, before discussing the possible solution to the drawback just presented, we must first look at 3D pose estimation. This is discussed next.

CHAPTER III

3D POSE ESTIMATION VIA STOCHASTIC DYNAMICS AND PARTICLE FILTERING

In this chapter, we present a particle filtering approach for pose estimation of two corresponding points sets. This chapter is based on [80, 81] and is organized as follows: In the next section, we discuss the particle filter and derive a novel local optimizer. In Section 3.2, we describe the registration algorithm along with the specifics of the prediction step, measurement model, and the resampling scheme. Section 3.3 provides numerical implementation details. Experimental results are given in Section 3.4. Finally, we conclude the chapter in Section 3.5.

3.1 Preliminaries

In this section, we derive a local optimizer for point set registration as well review some basic notions from the theory of particle filtering, which we will need in the sequel.

3.1.1 The Objective Functional

We now formulate and derive a novel local optimizer that is based on the correlation measure for point set registration. Specifically, we form an approximation for two sets of data described by mixture of Gaussian, and then show that the resulting registration scheme is a robust variant of ICP. We should note that although similarities exist with that of [95, 48], a key difference is that we keep explicit point correspondences. That is, rather than smoothing point sets in a constant or multi-scale fashion, we later employ this optimizer in the measurement functional as the "local" component in an otherwise "global" scheme.

3.1.1.1 An Approximation of the Correlation Measure for a Mixture of Gaussians

In what follows, we have assumed that we are given two mixtures of Gaussian distributions. We denote these distributions $m(y)$ and $d(y)$ as *model* and *data*, respectively. Specifically, they have the form $m(y) = \sum_{j=1}^{N_m} \alpha_j \phi_{m_j}(\mathbf{y}|\mu_{m_j}, \Sigma_{m_j})$ and $d(y) = \sum_{i=1}^{N_d} \beta_i \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i})$ where μ and Σ are the mean vector and covariance matrix of each mixture component, respectively.

Let us now assume that it is possible to obtain a closed-form expression for the correlation measure between these two mixtures of distributions, and that the modes of each of the neighboring mixtures are far apart. In other words, as in [37], one needs consider only a component of a mixture in evaluating $m(y)$ or $d(y)$. We note that although this assumption is generally valid for point sets, it may not be for other applications. Nevertheless, it can be seen that one appropriate approximation of the correlation measure is to match a single component of $m(y)$ to each component of $d(y)$, that is,

$$\begin{aligned} B(m,d) &= \sum_{i=1}^{N_d} \int \beta_i \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}) \cdot m(y) dy \\ &\approx \sum_{i=1}^{N_d} \max_j \left(\int \beta_i \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}) \cdot \alpha_j \phi_{m_j}(\mathbf{y}|\mu_{m_j}, \Sigma_{m_j}) dy \right) \\ &\approx \sum_{i=1}^{N_d} \max_j \left(B(\alpha_j \phi_{m_j}(\mathbf{y}|\mu_{m_j}, \Sigma_{m_j}), \beta_i \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i})) \right). \end{aligned}$$

Given the above assumptions, the term $\alpha_j \phi_{m_j}(\mathbf{y}|\mu_{m_j}, \Sigma_{m_j})$, which is in the same proximity of the component $\beta_i \cdot \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i})$, dominates the integral $\int \beta_i \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}) \cdot m(y) dy$. Now the term $\max_j(\cdot)$ above is realizable by computing the following:

$$B(m,d) \approx \sum_{i=1}^{N_d} \int \alpha_i^* \beta_i \phi_{m_i}^*(\mathbf{y}|\mu_{m_i}^*, \Sigma_{m_i}^*) \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}), \tag{11}$$

where α_i^* and $\phi_{m_i}^*(\mathbf{y}|\mu_{m_i^*}, \Sigma_{m_i}^*)$ correspond to the component that is the minimum Euclidean distance for the i^{th} component in the mixture modeled by $d(y)$.

3.1.1.2 Local Optimizer for Point Set Registration

Suppose now that we are given two point sets that lie in $\mathbb{R}^n$. Previously denoted as *model* and *data*, we can further describe each finite point sets by their respective elements $\{\mathbf{m}\}_{i=1}^{N_m}$ and $\{\mathbf{d}\}_{i=1}^{N_d}$. That is, each point within their respective point clouds forms a Gaussian distribution, whereby the mean vector is the location of a point and the covariance is the identity matrix, i.e., $\mu_{m_i} = m_i$, $\Sigma_{m_i} = I$. Assuming a rigid body transformation, $T(\vec{d}, \theta) : \mathbb{R}^n \mapsto \mathbb{R}^n$, for a set data points $\vec{d_i}$ and model points $\vec{m}_j$, we seek to find a rotational matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ that minimizes the following energy functional:

$$E = \sum_{i=1}^{N_d} \int \beta_i \alpha_i^* \cdot \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}) \phi_{m_i}^*(\mathbf{y}|\mathbf{R}\mu_{m_i}^* + \mathbf{t}, \mathbf{R}\Sigma_{m_i}^*\mathbf{R}^T). \tag{12}$$

Similarly to that of [48], we are now able to employ the following formula

$$\int \phi_1(\mathbf{y}|\mu_1, \Sigma_1)\phi_2(\mathbf{y}|\mu_2, \Sigma_2) = \phi(0|\mu_1 - \mu_2, \Sigma_1 + \Sigma_2). \tag{13}$$

Thus, we then have that

$$\begin{aligned}
E &= \sum_{i=1}^{N_d} \int \beta_i \alpha_i^* \cdot \phi_{d_i}(\mathbf{y}|\mu_{d_i}, \Sigma_{d_i}) \phi_{m_i}^*(\mathbf{y}|\mathbf{R}\mu_{m_i}^* + \mathbf{t}, \mathbf{R}\Sigma_{m_i}^*\mathbf{R}^T) \\
&= \sum_{i=1}^{N_d} \beta_i \alpha_i^* \cdot \phi(0|\mu_{d_i} - \mathbf{R}\mu_{m_i}^* - \mathbf{t}, \Sigma_{d_i} + \mathbf{R}\Sigma_{m_i}^*\mathbf{R}^T) \\
&= \lambda \sum_{i=1}^{N_d} \omega_i \exp\left(-\tfrac{1}{2}(\mathbf{R}\mu_{m_i}^* + \mathbf{t} - \mu_{d_i})^T(\Sigma_{d_i} + \mathbf{R}\Sigma_{m_i}^*\mathbf{R}^T)^{-1}(\mathbf{R}\mu_{m_i}^* + \mathbf{t} - \mu_{d_i})\right) \\
&= \lambda \sum_{i=1}^{N_d} \omega_i \exp\left(-\tfrac{1}{4}(\mathbf{R}m_i^* + \mathbf{t} - d_i)^T(\mathbf{R}m_i^* + \mathbf{t} - d_i)\right) \\
&= \lambda \sum_{i=1}^{N_d} \omega_i \exp\left(-\tfrac{1}{4}||d_i - \mathbf{R}m_i^* - \mathbf{t})||^2\right),
\end{aligned} \tag{14}$$

where $\omega_i = \beta_i \alpha_i^*$ and

$$\lambda = \frac{1}{(2\pi)^{N/2}|\Sigma_{d_i} + \mathbf{R}\Sigma_{m_i}^*\mathbf{R}^T|^{1/2}} = \frac{1}{(2\pi)^{N/2}2\sqrt{2}}.$$

Moreover, if we can obtain the surface normals of the *model* set, we can then further increase the robustness of the optimizer. This is done by dotting equation (14) with the (outward) surface unit normal $\vec{n}$ of the corresponding model point. The resulting expression that is to be optimized is now given as

$$E = \lambda \sum_{i=1}^{N_d} \omega_i \exp\left(-\tfrac{1}{4}||\mathbf{R}n_i^* \cdot [d_i - \mathbf{R}m_i - \mathbf{t})]||^2\right) \tag{15}$$

where $\vec{n}_i^*$ is the corresponding outward surface unit normal associated with model point m_i^*. Typically, a correspondence matrix C is used to relate the associated model point m_j to the data point d_i. After establishing these point correspondences, the optimal transformation $\phi = \{\vec{t}, R\}$ can be computed with the minimization of equation (15).

For the derivation of minimizing equation (15), the interested reader may refer to Appendix A. It is important to note though that after the transformation parameters are computed, a new "image" of the correspondence matrix is formed by applying the transformation $T(\vec{d}, \phi)$. The algorithm proceeds in an iterative manner until convergence or a stopping criterion is reached.

As compared to the methodology proposed in [48, 95], we have explicitly kept the point-to-point correspondences. Moreover, by keeping this explicit representation, we are then able to incorporate surface normals. The resulting optimizer can be seen as a robust variant of ICP, whereby we penalize outliers through the exponential term. Thus, we refer to this as a "local" optimizer in the sense of its narrow band of convergence. However, when used in conjunction with a particle filter, the basin of convergence is significantly widened. We also note that other optimizers [17, 31, 44] may be considered instead of the proposed functional presented here.

3.1.2 Particle Filtering

We now briefly revisit the basic notions and the generic setup of particle filtering as well as its motivation in point set registration.

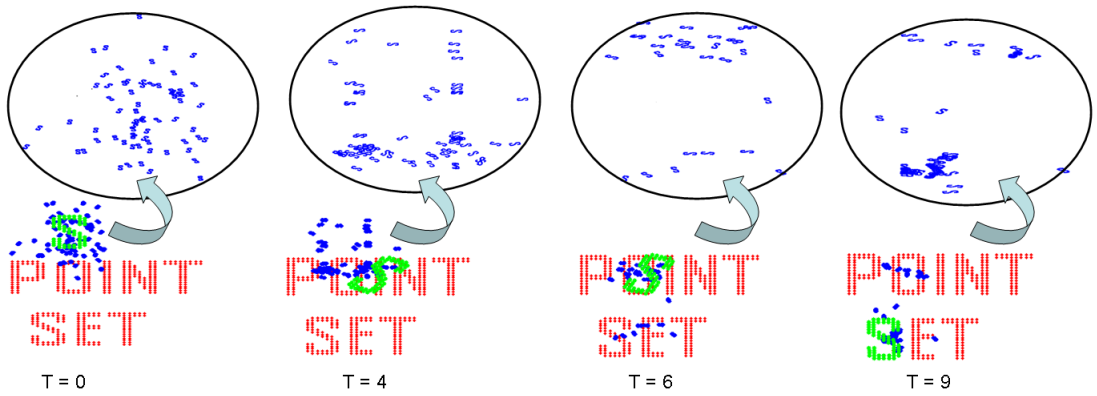


Figure 10: Illustration of several time steps of the proposed particle filtering approach. Sample pool of particles are shown for each step by a rescaled version of an oriented blue "S." The "Best Fit" Particle is shown as a green "S."

3.1.2.1 Background and Generic Scheme

Letting $x \in \mathbb{R}^n$, Monte Carlo methods allow for the evaluation of a multidimensional integral $I = \int g(x)dx$ via a factorization of the form $I = \int f(x)\pi(x)dx$, whereby $\pi(x)$ can be interpreted as a probability distribution. Taking samples from such a distribution in the limit yields the estimate of I that would otherwise be difficult or impossible to compute. However, generating samples from the posterior distribution is usually not possible. Thus, if one can only generate samples from a similar density $q(x)$, the problem becomes one of "importance sampling." That is, the Monte Carlo estimate of I can be computed by generating $N >> 1$ independent samples $\{x^i; i = 1, ..., N\}$ distributed according to $q(x)$ by forming the weighted sum: $I_N = \frac{1}{N}\sum_{i=1}^{N} f(x^i)w(x^i)$, where $w(x^i) = \frac{\pi(x^i)}{q(x^i)}$, represents the normalized importance weight. Consequently, by employing Monte Carlo methods in conjunction with Bayesian filtering, the authors in [39] first introduced the particle filter (PF). We refer the reader to [75, 29] for an in-depth discussion on Monte Carlo methods and particle filtering schemes.

Now considering $x_t \in \mathbb{R}^n$ to be a state vector with $z_t \in \mathbb{R}^n$ being its corresponding measurement, particle filtering is a technique for implementing a recursive Bayesian filter through Monte Carlo simulations. At each time t, a cloud of N particles is produced $\{x_t^i\}_{i=1}^N$, whose empirical measure closely "follows" $p(x_t|z_{0:t}) = \pi_t(x_t|z_{0:t})$,

the posterior distribution of the state given the past observations $z_{0:t}$.

The algorithm starts with sampling N times from the initial state distribution $\pi_0(x_0)$ in order to approximate it by $\pi_0^N(x_0) = \frac{1}{N}\sum_{i=1}^{N}\delta(x_0 - x_0^i)$, and then implements Bayesian recursion at each step. With the above formulation, the distribution of the state at $t-1$ is given by $\pi_{t-1}(x_{t-1}|z_{0:t-1}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(x_{t-1} - x_{t-1}^i)$. The algorithm then proceeds with a **prediction step** that draws N particles from the proposal density $q(x_t|z_{0:t-1})$. With appropriate importance weights assigned to each particle, the *prediction distribution* can now be formed in a similar fashion as above, i.e., $\hat{\pi}_t(x_t|z_{0:t-1}) = \frac{1}{N}\sum_{i=1}^{N} w_t^i \delta(\hat{x}_t - \hat{x}_t^i)$. Then, in the **update step**, new information arriving online at time t from the observation z_t is incorporated through the importance weights in the following manner:

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)}. \tag{16}$$

From the above weight update scheme, the *filtering distribution* is given by $\tilde{\pi}_t(x_t|z_{0:t}) = \frac{1}{N}\sum_{i=1}^{N} w_t^i \delta(x_t - x_t^i)$. Resampling N times with replacement from $\tilde{\pi}_t$ allows us to generate an empirical estimate of the posterior distribution π_t. Even though $\tilde{\pi}_t$ and π_t both approximate the posterior, resampling helps increase the sampling efficiency as particles with low weights are generally eliminated.

3.1.2.2 Registration as a Filtering Problem

Although much of the particle filtering work related to computer vision has involved the area of target estimation such as visual tracking [8, 73], the general framework is valid for any problem for which one desires the posterior distribution. We should note that while in the context of registration there exists no physical time t for which information is received online like that of tracking, most point set registration methodologies involve the estimate of a transformation through the establishment of correspondences that do change as a particular algorithm converges. To this end, we can induce an artificial time t where "information" can be regarded as point

correspondences for a given pose estimate. With this, we can then adopt the generic scheme presented in Section 3.1.2.1 for the purpose of point set registration where one would like to estimate the pose transformation in the posterior. Moreover, if we are interested in estimating the transformation given the correspondences, we do not need the complete paths of particles from time 0 to time t. Consequently, a translation prior can be employed as our proposal density. This is given as

$$q(x_t|x_{t-1}^i, z_k) = p(x_t|x_{t-1}^i). \tag{17}$$

Equation (17) means that our proposal density is dependent on the past state estimation. This very assumption is used in forming the prediction model of Section 3.2.2 from the "motion" alignment error. It is also a common choice for particle filtering applications, such as target tracking. In addition to assuming a translational prior, other design choices including the prediction model, the measurement functional, and the resampling scheme impact the algorithm's behavior. This will be the discussed next.

3.2 Proposed Framework

In this section, we cast the problem of pose estimation for point sets within a particle filtering framework. We will explicitly show that by modeling the uncertainty of the transformation, the resulting approach is substantially less prone to the problem of local minima, and the algorithm is robust to noise, clutter, and initialization. An overview can be found in Figure 10.

3.2.1 State Space Model

Point set registration can be viewed as a posterior estimation problem. That is, if we are given the correspondences at a specific time t, we then seek to predict the pose parameters that optimally aligns two point clouds.

Throughout the rest of this chapter, we assume that the registration problem is

restricted to 2D and 3D point sets. Specifically, we let $x_t \in \mathbb{R}^3$ and $x_t \in \mathbb{R}^6$ represent the respective state space of a rigid body transformation, i.e.

$$x(t) = \begin{pmatrix} \vec{t} \\ \vec{\theta} \end{pmatrix}(t). \tag{18}$$

For the 3D case, the translation and rotation vectors are $\vec{t} = [t_x, t_y, t_z]^T$ and $\vec{\theta} = [R_x, R_y, R_z]^T$, respectively. Similarly, for 2D point sets, the state space is $x(t) = \{t_x, t_y, \theta\}^T$. As stated previously, we exploit the uncertainty of the registration in our prediction step. This forms an estimate of the state $\hat{x}_t$ from the stochastic diffusion modeling of the distribution $p(x_t|x_{t-1}, z_{t-1})$. A detailed discussion is provided in Section 3.2.2, where it is also shown that the basis of this prediction model can be viewed as approximation to the selection of the proposal density. After an estimate is formed, we obtain an observation at time t, which is the "image" formed under the intermediate update of the correspondence matrix, $C(T(\vec{d}, \phi))$.

Thus, the observation space is given as follows:

$$z(t) = \begin{pmatrix} \vec{t}^m \\ \theta^m \end{pmatrix}(t) \tag{19}$$

where θ^m and $\vec{t}^m$ are the measured transformation parameters. In other words, the measured parameters are the optimal transformation estimate obtained from the local refinement in the observational functional (see Section 3.2.3). We should note that unlike the work of [56], our measurement functional is not just based on the explicit point correspondences. Instead, because we treat these correspondences as "information" that is received online as in the case of a video sequence in visual tracking [8], we are focused on measuring the transformation estimate given the correspondences. This key difference allows us to incorporate dynamics in the prediction model.

(a) (b)

Figure 11: Simplistic case of the uncertainty in point set registration. (a) For translation, parameter estimates are largest for t_x. (b) For rotation, the estimates are largest in R_x

3.2.2 Prediction Model

We seek a model for the prediction distribution, which can best describe uncertainty of the transformation during the registration process.

3.2.2.1 "Motion" Alignment Error

Inspired by [91], let us define the "motion" error for each particle $\{x^i; i = 1, ..., N\}$ that is learned online at time t as

$$e(x^i_{t-1}, \hat{x}^i_{t-1}) = x^i_{t-1} - \hat{x}^i_{t-1}, \tag{20}$$

where $\hat{x}_{t-1}$ and x_{t-1} are the predicted and measured state at $t-1$, respectively. Then the covariance of "motion" error is given as

$$S^i_{t-1} = E[e(x^i_{t-1}, \hat{x}^i_{t-1})e(x^i_{t-1}, \hat{x}^i_{t-1})^T]. \tag{21}$$

Assuming independence amongst the error parameters, S_{t-1} basically describes the variability or severity of motion in each of the principal axis for a rigid body transformation. This is shown in Figure 11. Here, a displacement is made for the pure translation and rotation case of a truck model. In these simplistic cases, the transformation estimate will be predominantly in the x-direction for (a) or about the rotational x-axis for (b).

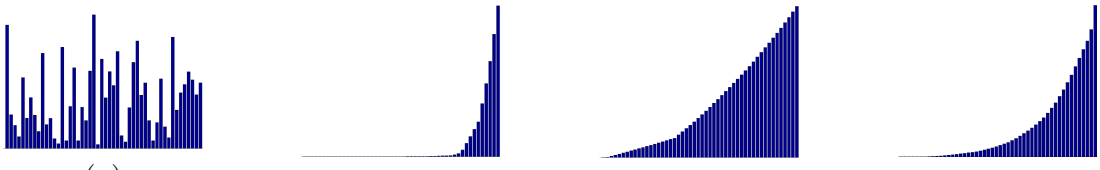


Figure 12: Viewing the Posterior Distribution and Effects of Gradient Descent on the Cumulative Distribution Function (CDF). (a) Typical Result of the Posterior Distribution in Point Set Registration (b) CDF exhibiting "Sample Degeneracy" (c) CDF exhibiting "Sample Impoverishment" (d) CDF exhibiting when choosing Optimal L.

By computing these transformation estimates from local variations in the pose parameters, we propose to explore the space described by their principal components of motion in a non-deterministic fashion. It is important to note that we *only* seek perturbations in the posterior using the objective functional. That is, we do not want to fully employ the optimizer discussed in the previous section, nor do we want to make empirical estimates from the correspondences alone obtained at time t. The consequence and implications of properly invoking the objective functional will be discussed in both the measurement functional as well as in our resampling scheme.

3.2.2.2 Proposal Density

Now if we assume a translational prior for the proposal density, we can then model the multivariate distribution with the use of Parzen estimators. This yields

$$\begin{aligned} q(x_t|x_{t-1}^i, z_k) &= p(x_t|x_{t-1}^i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{K}\left(\frac{x_t^i - x_{t-1}^i}{h_{t-1}^i}\right), \end{aligned} \tag{22}$$

where h_{t-1}^i is the bandwidth of the kernel. Specifically, we choose $\mathbf{K}(x_t^i, x_{t-1}^i, h_{t-1}^i)$ to be a Gaussian function, i.e.,

$$\mathbf{K}(x_t^i, x_{t-1}^i, h_{t-1}^i) = \frac{\exp\left(-\frac{1}{2}(x_t^i - x_{t-1}^i)^T (h_{t-1}^i)^{-1}(x_t^i - x_{t-1}^i)\right)}{(2\pi)^{N/2}|h_{t-1}^i|^{1/2}}.$$

Because the bandwidth h_{t-1}^i changes the dynamics in our framework, we denote it as the "weighted diffusion" of particles with a dependence on the covariance of the

alignment error S^i_{t-1}, diffusion weight γ_{t-1}, and process noise v_{t-1}. That is,

$$h^i_{t-1} = \underbrace{\gamma_{t-1} S^i_{t-1}}_{\text{online}} + \underbrace{v_{t-1}}_{\text{off-line}} . \tag{23}$$

In the general formulation above, we have incorporated information that is learned on-line (alignment error) and *a priori* information (process noise) learned off-line. However, in the experimental validation and unlike that of [56, 62], we have assumed the process noise to be minimal. Moreover, it can be viewed as a non-deterministic annealing schedule where the parameters are learned on-line. That is, as $t \to \infty$, the uncertainty embedded in the diffusion process is naturally reduced ($\sigma^2_{t-1} \to 0$) leading to convergence. The resulting proposal density is given as

$$q(x_t | x^i_{t-1}, z_k) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{K}\left(\frac{x^i_t - x^i_{t-1}}{\gamma_{t-1} \cdot S^i_{t-1}} \right). \tag{24}$$

As mentioned in Section 3.1, the selection of the proposal density is a critical issue in the design of any particle filter [75]. Equation (24) describes a prediction that diffuses stochastically in the direction of motion (through the bandwidth term) providing a temporally coherent solution in the context of point set registration.

A simplification of the weight update scheme can now be made by substituting equation (24) into equation (16) yielding:

$$w^i_t \propto w^i_{t-1} p(z_t | \hat{x}_t). \tag{25}$$

In the next section, we propose a measurement functional that allows us to compute $p(z_t|\hat{x}_t)$, and the weight updating scheme in equation (25).

3.2.3 Measurement Model

The measurement function, $z_t = h\big(\hat{x}_t, C(t)\big)$, where $\hat{x}_t$ is a seed point (corresponding to a transformed point set), and $C(t) = C\big(T(\vec{d}, \phi)\big)$ is the "image" that becomes available at time t, can be described as follows:

1. Run minimization of the functional (15) for L iterations for each of the $\hat{x}_t^i$: the choice of L depends upon the local optimizer and the method of minimization (e.g., gradient descent, Gauss-Newton). This results in a local exploration of both the transformation and the degree of misalignment existing between point sets. See Section 3.2.4 for details.

2. Compute an update of the importance weight by equation (25) by defining $p(z_t|\hat{x}_t) \triangleq e^{\sum_{i=1}^{N_d} \|n_i \cdot (m_i - T(\vec{d},\phi))\|^2}$.

3. Construct a cumulative distribution function from these importance weights. Using the generic method in [75], resample N times with replacement to generate N new samples.

4. Select the transformed point set with the minimum energy as the measurement. Update the path of transformation for each particle, which is used by equation (20) to describe the "motion" alignment error.

Note from our above considerations that the posterior distribution and transformation parameters can vary drastically depending on the set of correspondences obtained at each step. For example, Figure 12a shows the posterior distribution that characterizes the energy for each particle obtained from step 2 above. Hence, we must not model the distribution as unimodal. Thereby, this justifies the use of a mixture distribution to capture the wide variety of particle motions. Next, we discuss the resampling scheme, and the importance of doing gradient descent for L iterations.

3.2.4 Resampling Model

The resampling step is introduced into particle filtering schemes as a solution to "sampling degeneracy," which is unavoidable in sequential importance sampling. Indeed, the authors in [29] show that the variance of importance weights will in general only increase over time.

Moreover, in the context of point set registration, particles may not tend toward high likelihood regions of the posterior distribution if a general resampling scheme such as that of [75] is adopted. This is generally due to the empirical estimates that are formed after the prediction step. In this chapter, the motivation of doing gradient descent of L iterations of a chosen local optimizer is to explore the uncertainty in the correspondences and registration process. However, it can also be seen that a proper choice of L not only exploits the uncertainty as needed in the "motion" alignment error, but it also alleviates two extremes in the generic resampling scheme, "sample degeneracy" and "sample impoverishment." We discuss these two cases as well as how to properly choose L.

3.2.4.1 Choosing L too small: Sample Impoverishment

Although resampling attempts to solve "sampling degeneracy," it induces another problem known as "sample impoverishment." In other words, by not invoking the local optimizer within the measurement functional or by choosing L to be too small, particles will never tend toward the high likelihood region of the posterior. This is shown in Figure 12c. Here, a cumulative distribution shows that these weights are about equal, and the resampling step will not eliminate those particles that are regarded as "bad." This results in a poor approximation to the posterior distribution, and the registration may fail.

3.2.4.2 Choosing L too large: Sample Degeneracy

On the other hand, choosing L too large would effectively allow the particles to converge towards the local minima. This is not desirable since the state at t and $t-1$ would lose dependency. Indeed, this can be regarded as "sample degeneracy" as all the particles will tend toward one region during the resampling process. This can be seen in Figure 12b, where the cumulative distribution shows only few particles with a high-likelihood, while the majority have negligible impact.

3.2.4.3 Reasonable Choice of L

Thus, the choice of L should be chosen in a manner such that we avoid the two extremes mentioned above. In particular, the choice of L should produce the cumulative distribution similar to that of Figure 12d. In the present work, we have chosen L so that we are able to establish the notion of uncertainty as discussed in Section 3.2.2.1. However, in general, this also results in a resampling scheme that mitigates both "sample degeneracy" and "sample impoverishment." It also important to note that from a filtering perspective, *the choice of L depends on how much one trusts the system model versus the obtained measurement.* That is, if we choose L to be large, we completely depend on the measurement functional or local optimizer since this is run to convergence. On the other hand, if we choose L to be negligible, then the registration process is driven by process noise. Note, we have assumed the process noise to be small, but this can be easily incorporated in the current framework with minimal changes. In addition, it is also valuable to note the sensitivity of choosing L with respect to the performance of our particular filtering approach. Although one ideally would like to choose L so that the "uncertainty" to a particular problem is exploited, this may not always be the case. For our experiments, we have found that a choice $L = 7$ for gradient descent and $L = 3$ for Gauss-Newton's method have given robust results. However, values ranging from $L \in [2, 4]$ and $L \in [4, 12]$ for Gauss-Newton and gradient descent, respectively, have been used without significant performance loss. It should be noted that the choice of L also depends on the type of local optimizer used.

3.2.4.4 Dimensionality of State Space vs Number of Particles

Although we are focusing on rigid transformations pertaining to 2D and 3D point sets, it is noteworthy to mention the state parameter dimension in relation to the number of particles required. From a theoretical perspective, it is well known that the number of

particles grows exponentially with dimensionality of the state space [29, 75]. However, in practice we have found that our particular setup does not require an exponential growth in the particle size when increasing the state space dimension. This is due to the fact of invoking a gradient descent algorithm in the measurement functional. Moreover, this framework has a similar setup to [73], in which the authors proposed to explore an infinite dimensional space of curves through a minimal amount of samples (30 particles) as compared to the CONDENSATION filter (1200 particles) [8]. From this, it can be expected (without proof) that if we extend the proposed approach to non-rigid registration, the amount of samples should not drastically increase. Of course this is also dependent on the choice of optimizer that is employed within the measurement functional.

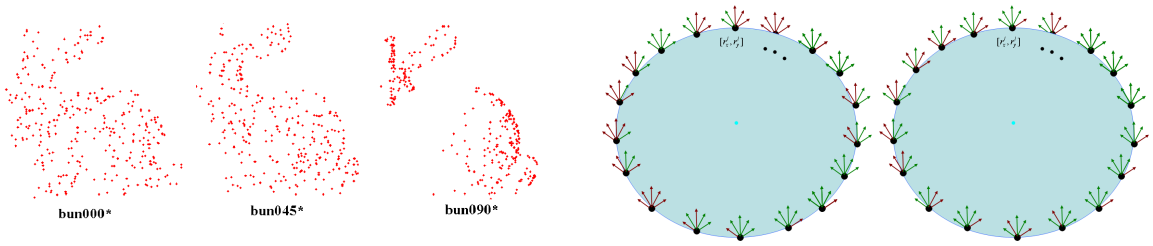
3.3 Implementation

Here, we provide implementation and numerical details of the algorithm described in Section 3.2.

3.3.1 Numerical Details

Experiments performed on both 2D and 3D data sets are implemented by minimizing the objective functional with the gradient descent approach. For a fast calculation of the correspondence matrix, we use a "KD tree" for the model points. Nearest neighbor searches are then easily performed for the varying data sets.

In addition, as with any particle filtering scheme, one needs to determine the number of particles and the initial population. In the present algorithm, we employ 100 particles. However, if empirical estimates are made in the posterior as in [56], the number of particles drastically increases. This is demonstrated in Section 3.4.3.1. Hence, this is yet another motivation for proposing the measurement function in Section 3.2.3. Assuming no prior knowledge of the specifics of the registration task, we adopt the following scheme for determining the initial population. We generate n



(a) (b) (c)

Figure 13: Illustrating the Domain of Convergence. a) Three 2D projected models derived from the Stanford Bunny data set. b) Convergence Results for ICP. c) Convergence Results for Proposed Optimizer. Note: Arrows are positioned at varying 20° increments and Red Arrows and Green Arrows denote "Failures" and "Success," respectively.

Gaussian distributed particles about the given initialization. Specifically, we apply a rigid transformation for each particle with a translational variance of $\vec{t} = \tau * \mu_{data}$, where τ is chosen from $[\frac{3}{10}, \frac{5}{10}]$ and μ_{data} is the computed range of the data point set. Similarly, the variance for rotational vector is given by randomly selecting an axis of rotation (i.e., R_x,R_y, or R_z) with an angle $\varphi = [\frac{\pi}{4}, \frac{\pi}{2}]$.

Lastly, we allow the algorithm to run to convergence for each of experiments performed in Section 3.4. As stated previously, our approach can be considered a global-to-local technique in which we non-deterministically anneal the perturbation of states at time t through dynamics. Consequently, the algorithm terminates once the mean diffusion of particles crosses a certain threshold criterion (e.g., $\frac{\sum_{i=1}^{N}(x_{t-1}^{i} - \hat{x}_{t-1}^{i})}{N} < \epsilon$, where ϵ is a user specified value). In our experiments, ϵ is $< 1°$ for rotational angle and $< .7$ for the translation vector. More importantly, our final estimate is compiled from the optimal transformation estimate obtained through the measurement functional. In other words, it is the "best fit" particle's transformation that is chosen.

3.4 Experimental Results

We provide both quantitative and qualitative experimental results for both 2D and 3D point sets that undergo a rigid body transformation. Because particle filtering can be regarded as a stochastic optimization process belonging to a class of random

sampling methods, quantitative experiments shown were tested over repeated trials. Consequently, we report both the gaussian average and standard deviation of these results so as to provide the reader with some notion of "success."[1]

We compare the proposed framework with a generic particle filtering algorithm similar to [56] as well the Kernel Correlation registration scheme of [95]. These specific tests demonstrate the robustness (and limitations) of the algorithm to **initialization**, **partial structures** (or clutter), **partial overlapping point sets**, and **noise**. Moreover, we provide experimental justification for employing **stochastic dynamics** in a filtering framework as opposed to purely using deterministic annealing to drive the registration process or performing a multiple hypothesis test. Lastly, a performance analysis (with respect to the number of particles used) along with the execution time and iteration count for each of the experiments is given in Section 3.4.4. However, before doing so, we must first validate the local optimizer proposed in Section 3.1.1.2 since it plays a crucial role the measurement functional of our filtering algorithm.

3.4.1 Domain of Convergence for Proposed Optimizer

In the first set of experiments, we validate the proposed local optimizer with our own implementation of ICP. Specifically, we use the bun045*, a projection of bun045 from the Stanford Bunny Set [1]. This can be seen in Figure 13a, and is discussed in more detail in Section 3.4.2.3. In a similar validation manner to that of [91], the projected point set was initialized at positions sampled on a circle whose radius is half the fixed shape width. Initial rotations of $\pm 40^\circ$ in 20° increments were tested. ICP results are shown in Figure 13b while our proposed optimizer convergence results are shown in Figure 13c. Through user visualization, each arrow marked red was considered a "failure" while green represented a "successful" registration. The proposed optimizer registered a total of 62 scans while ICP registered 49. We note that as an optimizer

[1] Unless explicitly stated, we consider an alignment successful, if the registration offset is $< 2^\circ$ and the norm of translation offset is < 2.

used in a stand-alone fashion, other methods such as [95, 19] may result in better transformation estimates.

3.4.2 Comparative 2D Rigid Registration

In the second set of experiments, we compare the Kernel Correlation (KC) approach [95] with our algorithm here. The MATLAB code of the KC algorithm is made available on the authors' website (http://www.cs.cmu.edu/ ytsin/KCReg/). In this algorithm, a global cost function is defined such that the method can be interpreted as a multiply-linked ICP approach. Rather than define a single pair of correspondences, one point set must interact with *each* point in the opposing set, thereby eliminating point correspondences altogether. Our algorithm can also be re-interpreted as a switching stochastic ICP approach where one point set interacts with only a handful of correspondences. It should be noted that we do not claim the KC methodology is inferior to the proposed approach. The experiments are performed to aid and highlight the significance of keeping explicit point correspondences while trying to widen the band of convergence.

3.4.2.1 Qualitative Comparison: Partial Structures (Letter Search)

In this example, we create the words "POINT SET," and off-set the letter S with a rather large pose transformation as seen in Figure 14(a). Running the KC algorithm and the proposed approach, we attempt to recover this transformation. The task of finding a letter within a set of words is a typical partial matching problem. We performed the KC method for several varying kernel bandwidths, and found that $\sigma_{KC} = 2.5$ provided the most successful result. This is shown in Figure 14(c). In particular, as one increases the bandwidth σ_{KC}, the algorithm tends to align distributions spatially, which makes it particularly ill-suited for partial matching. The result of the particle filtering approach (number of particles is 100 with $L = 7$) described in this chapter is shown in Figure 14(e). The transformation is recovered.

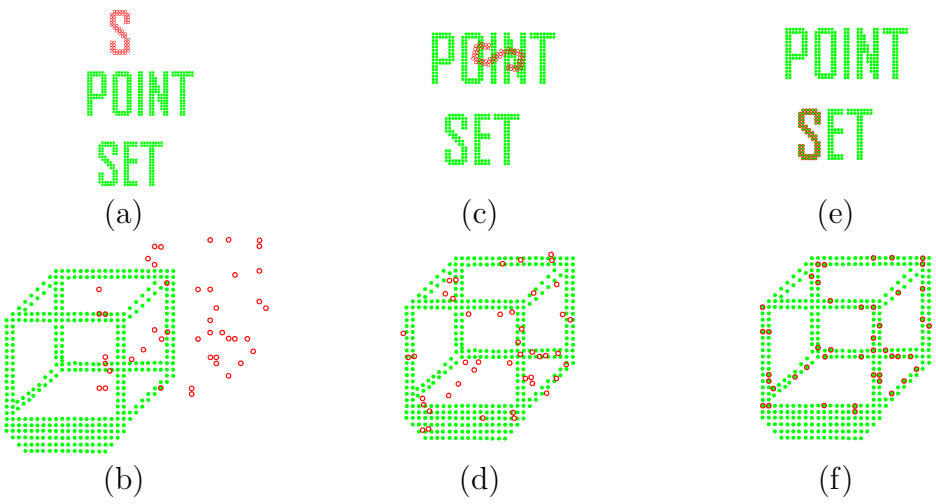


Figure 14: Examples of estimating the pose with points sets having clutter or sparseness. (a) Initial letter off-set. (b) Initial cube off-set. (c) KC word search result. (d) KC cube result. (e) Particle filtering word search result. (f) Particle filtering cube result.

3.4.2.2 Qualitative Comparison: Geometric Assumptions (Cube)

The next experiment deals with the case of differing densities across the two point sets. We should note that many point set registration algorithms make some tacit assumptions on the point set density. In another words, they assume point sets that have a similar density or geometry around a local neighborhood for each point within their respective sets. We refer the reader again to Figure 2(b) and (c) for an illustration of differing densities. In particular, the KC algorithm uses kernel density estimates to describe the (dis)similarity between points across the two sets. To overcome poor initialization, noise, or clutter, the kernel bandwidth must be increased. However, in doing so, the kernel smoothes the point sets, which makes it increasingly difficult to discriminate among individual points when working with sparse and dense sets. To demonstrate this, we generate 50 points from the model cube, which is itself composed of 400 points. A transformation $T(\vec{d}, \phi)$ is then applied to the extracted data set. Similar to the preceding section, we tested several kernel bandwidths, and found $\sigma_{KC} = 3$ to be the optimal choice. The result is shown in

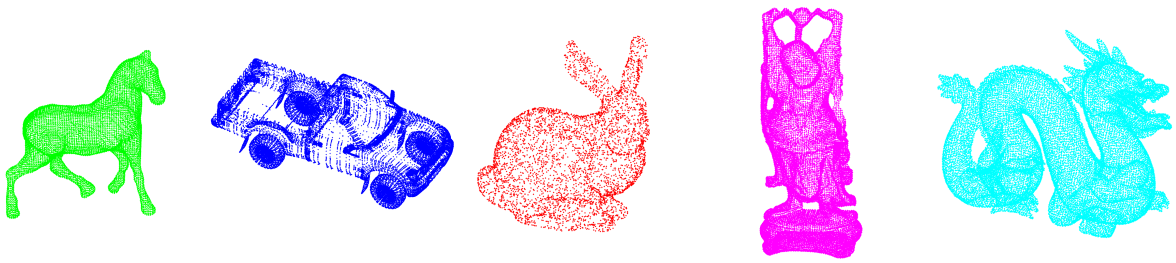


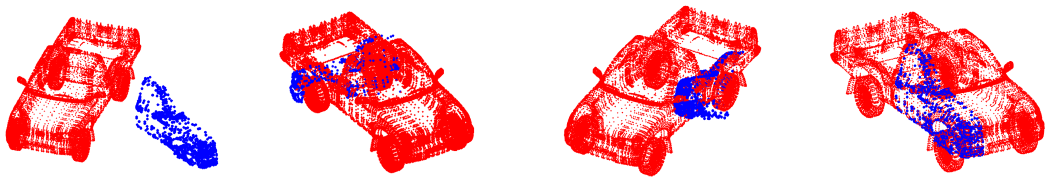
Figure 15: Different views of the 3D point sets used in this chapter.

Figure 14(d), where a suboptimal registration is obtained. A successful alignment is recovered using the proposed method (number of particles = 100, $L = 7$) as seen in Figure 14(f).

3.4.2.3 Quantitative Comparison: Initialization and Noise (2D Projections of Stanford Bunny)

In this experiment, we quantitatively compare the KC algorithm to our proposed particle filtering approach under noise and initialization. Because the 3D KC algorithm was not available online, and real life 2D point sets were not readily available, we opted to form 2D projections from three scaled models of the standard Stanford Bunny data set [1]. This is shown in Figure 13a. We note that while depth information is removed from the original model, the projection itself still represents the differing sampling and overlapping regions of each 3D Bunny model, making it suitable for a quantitative comparison. To further ensure validity of these projected scans, we performed supervised registration and found that a "success" or global minima is achieved if $\|\vec{t}\| < 2.5$ with a rotational offset $\vec{\theta} < 7^\circ$ for each scan pair.

Taking bun045*, bun090*, and bun090*, we formed a 100 possible combination pairs. We then applied a rigid transformation for each pair. In particular, we generate translations $\vec{t} = [t_x, t_y]^T$ from a normal distribution with each component having a standard deviation of 3, i.e. $\mathcal{N}(0, 3)$. A rotational offset was chosen from a uniform distribution $\mathcal{U}(0, \frac{\pi}{2})$. After a transformation is applied, the data set is sampled with replacement with Gaussian zero mean noise. The applied noise is $\mathcal{N}(0, 15)$. In this



(a) (b) (c) (d)

Figure 16: The importance of stochastic motion. (a) Initialization. (c) Result with deterministic annealing model. (c) Result with no dynamical model. (d) Result with stochastic motion model.

experiment, we generated noise levels of 0, 5, 10, and 25 percent substitution, and then we performed tests at each of the several varying noise levels. Further, the number of particles used is 100 with $L = 7$.

Table 2 shows the number of successful alignments for the KC algorithm compared under three different choices of a smoothing kernel with that of proposed approach. Specifically, we found that the kernel choice of σ_{KC}=3 to be optimal. Moreover, we repeated each trial 100 times to ensure the repeatability success of our algorithm (given that it is a random sampling technique). Interestingly, the proposed approach outperformed the KC algorithm for noise levels of 0, 5, 10 percent. However, the results were similar for 25 percent, and at times KC outperformed the proposed approach for certain kernel sizes. This can explained by the KC algorithm's ability to align centers of mass including the noise. Although this can be a limitation of using explicit point correspondences, aligning centers of mass may not be desirable if partial structures exist as shown in Section 3.4.3.1.

3.4.3 Rigid Registration of 3D Point Sets

Our experiments with 3D rigid registration of point sets use several 3D models as seen in Figure 15. Specifically, the Stanford Bunny, the Buddha, and the Dragon, are models obtained from the Stanford 3D Repository [1]. In addition to these models, several scans of a room, which first appear in [58], have also been used. While some of the partial scans do not contain the surface normals, we use the methodology proposed

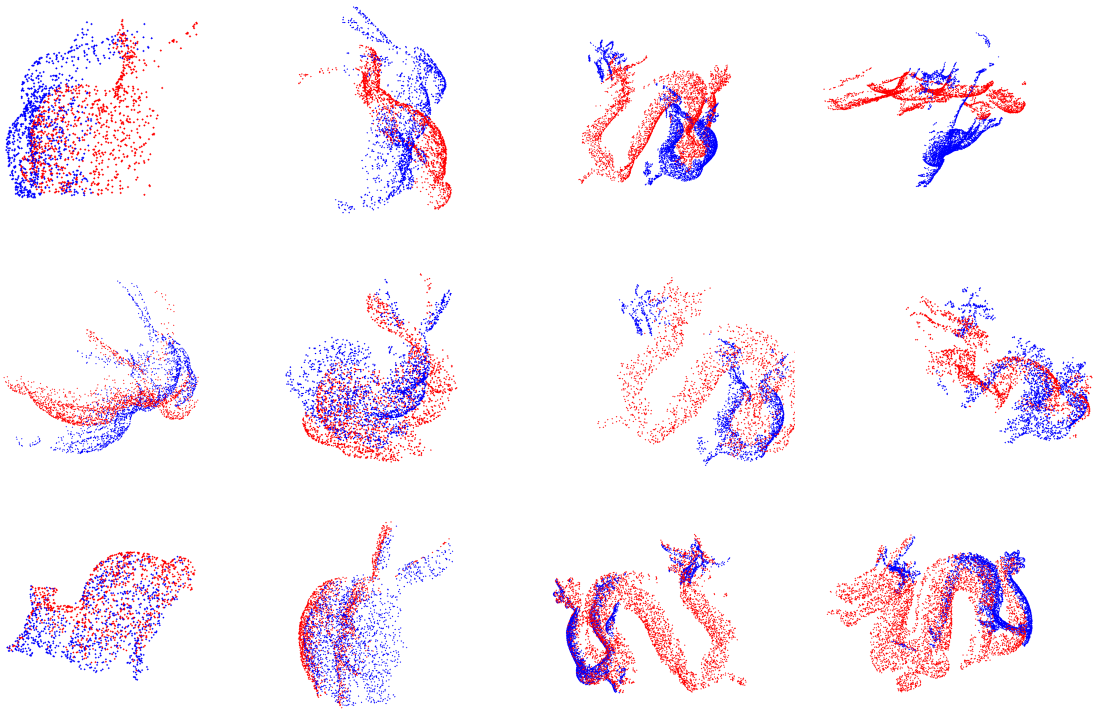
Noise	K.C. Alg. (σ_{KC}= 1)	K.C. Alg. (σ_{KC}= 3)	K.C. Alg. (σ_{KC}= 5)	P.F. (100 Trials)
0 %.	20	26	16	$\mu = 50.10$ $\sigma = 3.53$ max = 59 min = 41
5 %	15	16	13	$\mu = 31.94$ $\sigma = 3.75$ max = 41 min = 22
10 %	17	17	12	$\mu = 27.41$ $\sigma = 3.59$ max = 35 min = 16
25 %	16	18	20	$\mu = 18.45$ $\sigma = 3.34$ max = 27 min = 10

Table 2: 2D Comparative Analysis with Kernel Correlation Under Varying Levels of Noise and Initialization. Number of "Successful" Alignments are shown, where "Success" is denoted if $\|\vec{t}\| < 2.5$ with a rotational offset $\vec{\theta} < 7^{\circ}$ is found for each scan pair.

by [27]. The main focus here is the importance of stochastic motion dynamics and the algorithm's inherent robustness to noise and initialization as well as being to handle partial overlapping scans.

3.4.3.1 Quantitative Comparison: Motion Dynamics (Truck)

In this experiment, we demonstrate the importance of stochastic motion with our own implementation of a filtering scheme similar to that of [56] Specifically, we employ deterministic annealing for our process and measurement noise. Moreover, we replace the ICP functional with our optimizer so as to mitigate any problems caused from the objective functional as well as to highlight the differences between each filtering setup. Lastly, we also compare the above algorithms to a multiple hypothesis based testing technique (i.e., no dynamics in the current framework). One of the underlying

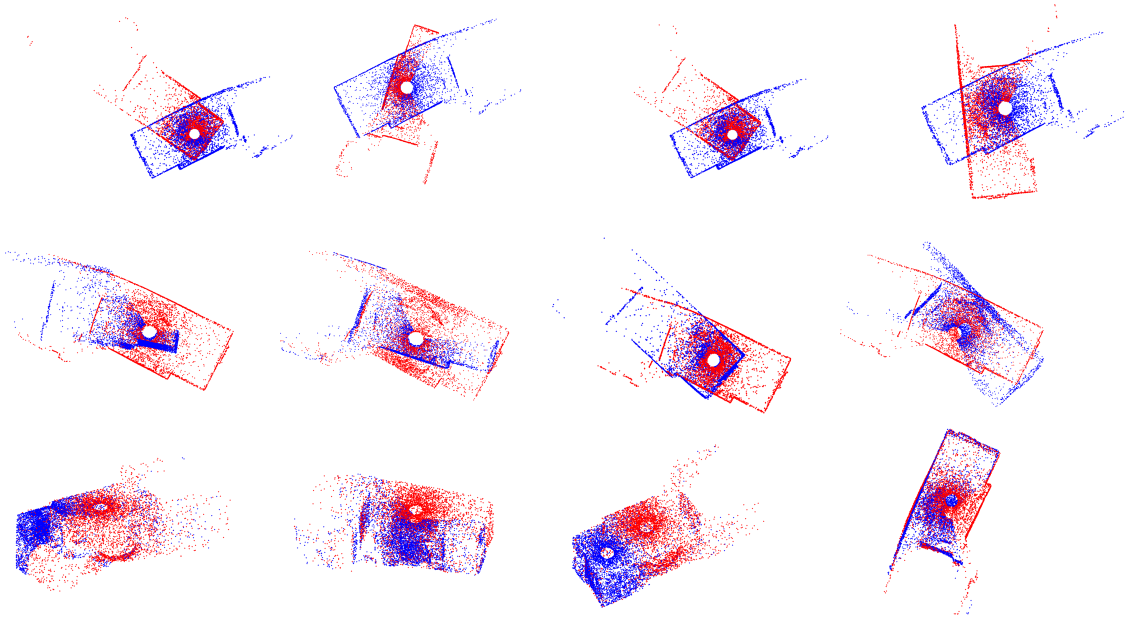


(a) bun000 to bun315 (b) bun045 to bun315 (c) drag72 to drag120 (d) drag0 to drag96

Figure 17: Partial Scans of an Apartment Room. Note the severe initial alignments. Top Row: Initializations. Middle Row: Intermediate Result of Proposed Approach Bottom Row: Final Result with Proposed Algorithm.

contributions in this chapter is to demonstrate that dynamics have a significant impact on the registration results. For example, in the case of 3D LADAR imagery [2], pose tracking algorithms involve segmentation of an object from a scene, which is then followed by point set registration. However, at a given instance of time, the object maybe only partially seen and segmented. This results in the task of partial matching in the context of registration. More importantly, when facing occlusions or erratic behavior, the extracted point set can be inaccurately initialized.

To demonstrate this, we first extract a cloud of points from the front side of the truck as seen in Figure 16(a), and center it with respect to the model. We then create 50 transformations. First, translations $\vec{t} = [t_x, t_y, t_z]$ are generated from a normal distribution with each component having a standard deviation of half the range of the model. The rotation angle θ is then chosen randomly along the z rotation axis, but



(a) apt10 to apt3 (b) apt10 to apt4 (c) apt10 to apt9 (d) apt10 to apt7

Figure 18: Partial Scans of an Apartment Room. Note the severe initial alignments. Top Row: Initializations. Middle Row: Intermediate Result of Proposed Approach Bottom Row: Final Result with Proposed Algorithm.

from a uniform distribution $\mathcal{U}(0, \frac{\pi}{3})$. In our comparison, we initialized the process and measurement noise as $.7*[\max(model) - \min(model)]$ and $.3*[\max(data) - \min(data)]$, respectively. The annealing rates were chosen to be .85 for the process noise and .7 for the measurement noise. Initial distributions were the same for each filtering setup.

For the case of a multiple hypothesis testing, the algorithm was able to register 13 scans. We ran our own implementation similar to that of [56] multiple times (trials = 50) for the 50 transformations. Using 1000 particles for the filter driven by process and measurement noise, we report a mean success rate of 25 and a standard deviation of 3.41 (over 10 trials). For the filtering method proposed in this chapter, which used only 100 particles, we report a mean success rate of 40.96 and a standard deviation of 1.91. The reasoning for such an improvement between the two particle filters can be explained by the fact that even with 1000 particles, at times "good" particles were

Noise	Optimizer ($\theta < 60°$)	P.F. (50 Trials). ($\theta < 60°$)	Optimizer ($\theta < 120°$)	P.F. (50 Trials) ($\theta < 120°$)
5 %.	83	$\mu = 91.17$ $\sigma = 1.62$ max = 94 min = 88	57	$\mu = 75.59$ $\sigma = 2.81$ max = 83 min = 72
10 %.	86	$\mu = 92.89$ $\sigma = 1.74$ max = 97 min = 90	55	$\mu = 74.23$ $\sigma = 3.40$ max = 80 min = 66
25 %.	79	$\mu = 85.20$ $\sigma = 1.96$ max = 88 min = 81	49	$\mu = 69.44$ $\sigma = 2.80$ max = 73 min = 63
35 %.	74	$\mu = 82.86$ $\sigma = 2.35$ max = 86 min = 78	44	$\mu = 66.44$ $\sigma = 2.16$ max = 71 min = 60

Table 3: 3D Performance Gain of Employing Proposed Particle Filtering Method in Conjunction with Proposed Local Optimizer Under Varying Levels of Noise and Initialization. Number of "Successful" Alignments are shown, where "Success" is denoted if $\|\vec{t}\| < 2$ with a rotational offset $\vec{\theta} < 2°$ is found for each scan pair.

driven away from the global minima by the high noise. In the same respect, the deterministic noise also allowed for "bad" particles to diffuse to the correct global minima. Figure 16 shows an example of when the proposed approach outperforms deterministic annealing and a multiple hypothesis testing.

3.4.3.2 Qualitative Results: Partial Non-Overlapping Data Sets (Bunny, Room)

An important task for many point set registration algorithms is the ability to properly register scans that exhibit only a partial overlap of the surface. In addition, depending on the acquisition of the point sets, the sampling of points may create inconsistences in the correspondences across two sets of interest. In what follows, we use L = 7 with 100 particles.

We begin with the famed 3D models from Stanford Repository, which are composed of models whose scans are taken in succession. We have limited our registration experiment to those scans that exhibit a poor initialization or poor overlaps. In particular, Figure 17 shows the successful registration of bun000 to bun045, bun000 to bun315, bud0 to bud336, and drag0 to drag96.

While Figure 17 demonstrates the notion of a non-overlapping surface, the scans are generally tested with "local" optimizers. Thus, we extend this experiment to scans of a room taken by a DeltaSphere$^{\text{TM}}$-3000 laser scanner. In particular, the large rotational offset along with sampling density and partial structures creates difficult problems for point set registration In Figure 18, we show the successful registration of four scans of a room.

3.4.3.3 Quantitative Results: Performance Gain of Employing P.F. Algorithm (Horse)

In this example, we extensively test the algorithm's performance in the case of large misalignment and large levels of noise. Moreover, we demonstrate the performance gain of employing our proposed particle filtering algorithm. First, we generate two series of 100 random transformations and apply them to a data set that is uniformly sampled from the model. The first series of transformations focuses on the case for which iterative techniques are commonly tested. In particular, we generate translations $\vec{t} = [t_x, t_y, t_z]$ from a normal distribution with each component having a standard deviation of 30, i.e. $\mathcal{N}(0, 30)$. This value is chosen according to the range of model points, $([-36, 33], [-67, 74], [-75, 82])$. The rotation angle θ is then chosen randomly along the z rotation axis, but from a uniform distribution $\mathcal{U}(0, \frac{\pi}{3})$. The second series of transformations is similar to the first set, except now the rotation angle is chosen from a uniform distribution $\mathcal{U}(\frac{\pi}{3}, \frac{2\pi}{3})$.

After a transformation is applied, the data set is sampled with replacement with Gaussian zero mean noise. The applied noise is $\mathcal{N}(0, 45)$, which is again chosen with

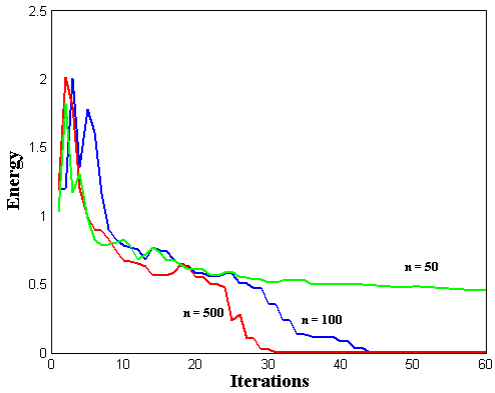


Figure 19: Plot of the energy convergence of the proposed approach with respect to the iteration count for three different sample sizes.

respect to the dimensions of the horse. In this experiment, we generated noise levels of 5, 10, 25, and 35 percent substitution, and then we performed tests at each of the several varying noise levels. Further, the number of particles used is 100 with $L = 7$, and the initial distribution has the same spread for each random transformation.

Table 3 above presents the number of successful alignments of running just the optimizer presented in Section 3.1.1.2 as well as the particle filtering algorithm proposed in this chapter. Because of the random nature of particle filtering, the experiment was repeated over 50 trials with the same initializations. We report both the mean success rate, standard deviation, as well as the minimum and maximum successful alignments for each case. Interestingly enough, the particle filtering approach outperforms the iterative-based technique even in the case where the rotational angle is not extreme. One can then see the significant improvement in widening the narrow band of convergence for the particular case of large off-sets and noise.

3.4.4 Performance Analysis

The final experiment examines the time-performance analysis of our particle filtering approach. Specifically, we focus on the breakdown limit of the number of particles needed for accurate registration in the case of the experiment performed in Section 3.4.3.1. We ran this experiment with the same transformations, but with the

additional particle sizes of 50 and 500. To ensure a notion of repeatability of success the number of trials was chosen to be 50. For the case of 500 particles, we report a mean success rate of 48, with a standard deviation of 1.15. Compared to 100 particles as previously tried, which we reported to have a mean success rate of 40.96 and standard deviation of 1.91, adding more particles increases the rate of success. In contrast, the "breakdown" limit was seen to be roughly 50 particles in which the algorithm's efficiency decreases to a mean success rate of 30.16 with a standard deviation of 2.24. This can be seen for a challenging initialization in Figure 19. In particular, a higher level of particles also enables the algorithm to converge at a slightly quicker rate.

Lastly, we report the execution time, iteration count and other relevant information for each of the experiments performed in this chapter in Table 4. It should be noted that our implementation of both the "local" optimizer and particle filtering algorithm were done in MATLAB v7.1 on a Intel Dual Core 2.66GHz with 4 GB memory. Also, in relative terms, our implementation of the filtering algorithm using deterministic annealing averaged 1228 sec over 50 iterations. This reduction in computational speed is due to nearest neighbor searches and unoptimized code. One final key note about the performance of our algorithm is its ability in reducing the sample size of particles. From this, both ICP and particle filtering can be currently parallelized onboard a tracking system in a much more efficient manner, where limitations occur in the need for resampling component of the particle filter.

Experiment	Model Dimension	Number of Model Points	Number of Data Points	Avg. Number of Iterations	Time (in Sec)
P.F. Gain Performance (Horse)	([-36,33],[-67,74],[-75,82])	1000	1000	15	18.3
Motion Dynamics (Tacoma Truck)	([-33.31,33.31],[-74.97,74.99],[-28.15,28.15])	1000	10000	36	64.60
2D Noise Comparison (Projected Bunny)	≈([-14,13],[-12,17])	299	299	6	5.31
2D Partial Structures (Letter Search)	([-16.2 20.8],[-24.32,15.68])	450	64	9	5.03
2D Geometric Assumptions (Cube)	([58.04 30.98],[-56.29,-26.95])	383	50	7	3.17
Partial Non-Overlap (bun045,bud96,..)	≈([-23, 31],[12 70],(-16 35])	1000	1000	48	40.93
Partial Non-Overlap (apt3,apt10,..)	≈([-222 141],[-97,135],[-65,41])	5000	5000	26	88.1

Table 4: Execution Time of Point Set Registration Algorithm

3.5 Chapter Conclusion

In this chapter, we cast the problem of pose estimation for point sets within a particle filtering framework that exploits the underlying variability in the registration process. This is done by estimating "motion" or uncertainty as local variations in pose parameters in the posterior. From this, a novel local optimizer based on the correlation measure is proposed and derived. Unlike [56, 62], the method does not require an annealing schedule and drives the registration with information that is learned online through a stochastic diffusion model. As compared to the KC algorithm [95], our approach only considers a set of correspondences in a switching like fashion. This enables the algorithm to correctly align point sets when dealing with partial structures or with the matching of dense and sparse sets.

Although we have presented a novel pose estimation scheme with the point registration algorithm, we note that there still exists major drawbacks that are inherent in formulating the solution. In particular, the proposed registration algorithm requires not only a costly stochastic optimization method like that of particle filtering, it also depends on point correspondences. Of course, having correspondences within the context of point set registration can be seen as a trade-off, we note that additional difficulties arise when these correspondences are generated from a 2D image so that they correspond to features on a 3D object. In a similar manner and as previously mentioned, segmentation also may not yield the desirable result due to the assumption of separable statistics. Thus, ideally we would like to combine the strengths of each algorithm with the intent of circumventing typical and sometimes inherent weakness of the respective algorithms. This unification of methodologies is presented next.

CHAPTER IV

UNIFYING 2D IMAGE SEGMENTATION AND 3D POSE ESTIMATION FOR A CLASS OF 3D OBJECTS

In this chapter, we present a non-rigid approach to jointly solve the tasks of 2D-3D pose estimation and 2D image segmentation. This chapter is based on [85, 82] and is organized as follows. In the next section, we begin with a generalization of the gradient flow of [24] for an arbitrary set of finite parameters. We then provide details for evolving both the shape parameters, which are obtained from performing Kernel Principle Component Analysis (KPCA) on a collection of 3D shapes[1], and the corresponding pose of an object. In Section 4.4, we present experimental results that highlight the robustness of the technique to noise, clutter, and occlusions as well as the ability to segment a novel shape that is not apart of the specified training set.

4.1 Kernel Principal Component Analysis (KPCA) Review

In this section, we review the fundamental concepts associated with kernel PCA as well as the pre-image approximation used, which we will need in the sequel.

4.1.1 KPCA Formulation

Let us begin with a set of data points, $\{X_1, X_2, ..., X_N\}$ that are members of the input space $\mathcal{M} \in \mathbb{R}^n$. This set can be mapped to a possibly higher dimensional feature space, denoted by $\mathcal{H}$, via the nonlinear map $\varphi : \mathbb{R}^n \mapsto \mathcal{H}$. Moreover, this map does not need to be explicitly known. Indeed, one can introduce a Mercer kernel,

[1]We assume as with many machine learning techniques that we have a catalog of 3D shapes describing a particular object. Specifically, one can use stereo reconstruction methods [101] or range scanners to obtain accurate models as shown in Figure 20. From this, we derive a variational approach to perform the task of non-rigid 3D pose estimation and 2D image segmentation.

Figure 20: Three 3D Training Sets used in this work. *Top Row:* Different 3D models of commonly used tea cups (6 of the 12 models used for the training are shown). *Middle Row:* Different 3D models of the number 4 (6 of the 16 models used for the training are shown). *Bottom Row:* Different 3D models of commonly seen helicopters (6 of the 12 models used for the training are shown)

which is defined to be a function $k(X_a, X_b)$ such that for all data points X_i, the kernel matrix

$$\mathbf{K} = \begin{pmatrix} k(X_1, X_1) & k(X_1, X_2) & \dots & k(X_1, X_N) \\ k(X_2, X_1) & \ddots & & \\ \vdots & & k(X_i, X_j) & \\ k(X_N, X_1) & & & k(X_N, X_N) \end{pmatrix}$$

is symmetric positive [61, 87]. According to Mercer's Theorem [60], computing $k(X_a, X_b)$ as a function of $\mathcal{M} \times \mathcal{M}$, amounts to computing the inner scalar product in $\mathcal{H}$: $k(X_a, X_b) = \langle \varphi(X_a), \varphi(X_b) \rangle$, with $(X_a, X_b) \in \mathcal{M} \times \mathcal{M}$. This scalar product in $\mathcal{H}$ defines a distance $d_{\mathcal{H}}$. For example, the L_2 distance is $d^2_{\mathcal{H}}(\varphi(X_a), \varphi(X_b)) = \|\varphi(X_a) - \varphi(X_b)\|^2 = k(X_a, X_a) - 2k(X_a, X_b) + k(X_b, X_b)$.

If we now consider the input to be a collection of shapes, one can then perform the KPCA method as presented in [61]. Let $\mathcal{T} = \{X_1, X_2, ..., X_N\}$ be a set of training

data. The centered kernel matrix $\tilde{\mathbf{K}}$ corresponding to $\mathcal{T}$, is defined as

$$\begin{aligned}\tilde{\mathbf{K}}(i,j) &= \langle(\varphi(X_i) - \bar{\varphi}), (\varphi(X_j) - \bar{\varphi})\rangle \\ &= \langle\tilde{\varphi}(X_i), \tilde{\varphi}(X_j)\rangle = \tilde{k}(X_i, X_j), \text{for}(i,j) \in [|1, N|] \end{aligned} \tag{26}$$

with

$$\bar{\varphi} = \frac{1}{N}\sum_{i=1}^{N}\varphi(X_i)$$

where

$$\tilde{\varphi}(X_i) = \varphi(X_i) - \bar{\varphi}.$$

In addition, since $\tilde{\mathbf{K}}$ is symmetric, it can be decomposed as

$$\tilde{\mathbf{K}} = USU^T, \tag{27}$$

where $\mathbf{S} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N)$ is a diagonal matrix containing the eigenvalues of $\tilde{\mathbf{K}}$. $\mathbf{U}$ $= [u_1, u_2, ..., u_N]$ is an orthonormal matrix, where the columns $u_i = [u_{i1}, u_{i2}, ..., u_{iN}]^T$ are the eigenvectors corresponding to the eigenvalues λ_i's. Furthermore, it can be shown that

$$\tilde{\mathbf{K}} = \mathbf{HKH} \tag{28}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{l}\mathbf{l}^T$, $\mathbf{l} = [1, 1, ..., 1]^T$ is a $N \times 1$ vector, and $\mathbf{I}$ is a $N \times N$ identity matrix.

Let $\mathbf{C}$ denote the covariance matrix of the elements of the training set mapped by $\tilde{\varphi}$. Then the n^{th} (orthonormal) eigenvector of $\mathbf{C}$ in the feature space is given as follows

$$V_n = \sum_{i=1}^{N}\frac{u_{ni}}{\sqrt{\lambda_n}}\,\tilde{\varphi}(X_i) = \frac{1}{\sqrt{\lambda_n}}\tilde{\phi}u_n, \tag{29}$$

where $\tilde{\phi} = [\tilde{\varphi}(X_1), \tilde{\varphi}(X_2), ..., \tilde{\varphi}(X_N)]$. That is, $\tilde{\phi} \in \mathbb{R}^{nXN}$ The subspace of the feature space $\mathcal{H}$ will be referred to as the *kernel PCA space*.

Now let X be *any* element of the input space $\mathcal{M}$. The projection of X on the kernel PCA space, spanned by the first l eigenvectors of $\mathbf{C}$, is then given by

$$P^l\varphi(X) = \sum_{n=1}^{l}\omega_n V_n + \bar{\varphi} \tag{30}$$

with $\omega = [\omega_1, \omega_2, ..., \omega_l]$ being the KPCA shape weights or coefficients.

We note that while the above formulation allows us to construct the projection from a linear combination involving the shape coefficients, it is only valid in the feature space. That is, given a test point $\mathbf{x} \in \mathcal{M}$, we would ideally like to compute its projection in the image space $\hat{\mathbf{x}} = \varphi^{-1}(P^l\varphi(\mathbf{x}))$. Unfortunately, because the map $\varphi : \mathbb{R}^n \mapsto \mathcal{H}$ is unknown, one can not directly obtain this projection. This is referred to in the literature as the *pre-image problem* [54]. In this chapter, we use the non-iterative pre-image result of [22] to directly evolve the KPCA coefficients. However, before doing so, we first present a few popular kernels and their corresponding pre-image result used in literature for shape-based learning.

4.1.2 KPCA Kernels

We now present several kernels that allows one to perform learning of shapes using linear and nonlinear PCA.

4.1.2.1 Linear PCA

In [55, 94], a method is presented to learn shape variations by employing PCA on a training set of shapes (closed curves) represented as the zero level sets of signed distance functions (SDF). In order to perform PCA, let us begin with with the polynomial kernel. This is given by

$$k_{\varphi_P}(\psi_i, \psi_j) = (c + \langle \psi_i, \psi_j \rangle)^d \tag{31}$$

where c is any constant, d is the degree (odd) of the polynomial, and ψ_i is the SDF associated with a shape in one's training set. Then by choosing $c = 0$ and $d = 1$, we arrive at the following kernel used to perform classical PCA on SDFs:

$$k_{id}(\psi_i, \psi_j) = \langle \psi_i, \psi_j \rangle = \int\int\int \psi_i(u, v, k)\psi_j(u, v, k)dudvdk \tag{32}$$

for the SDF's ψ_i and ψ_j: $\mathbb{R}^3 \mapsto \mathbb{R}$. The subscript id stands for the identity function: when performing linear PCA, the kernel used is the inner scalar product in the input

space. Hence, the corresponding mapping function $\varphi = id$. One should note that this latter integral is infinite if there is no bound. However, since this is being used in a shape learning context (PCA), there is an assumption on the size of the shapes within the given training set that gives the necessary boundedness and finiteness result. This certainly affects the distance, and makes it non-intrinsic.

4.1.2.2 NonLinear PCA

Choosing various types of nonlinear kernel functions $k(X_a, X_b)$ is the basis of nonlinear PCA. The exponential kernel has been a popular choice in the machine learning community and has proven to nicely extract nonlinear structures from data sets; see e.g. [61]. Using SDFs for representing shapes, this kernel is given by

$$k_{\varphi_\sigma}(\psi_i, \psi_j) = e^{\frac{-\|\psi_i - \psi_j\|^2}{2\sigma^2}} \tag{33}$$

where σ^2 is a variance parameter estimated *a priori* and $\|\psi_i - \psi_j\|^2$ is the squared L_2-distance between two SDF ψ_i and ψ_j. The subscript φ_σ stands for the nonlinear mapping corresponding to the exponential kernel. This mapping also depends on the choice of σ.

4.1.3 Pre-Image Approximation

In this section, we revisit the closed-form pre-image approximation proposed by [22, 71]. It can be seen that although the φ map is not necessarily known, we ideally would like to reconstruct the pre-image $\hat{\mathbf{x}}$ of a corresponding test point $\mathbf{x} \in \mathcal{M}$ such that the distance between the feature point $\varphi(\hat{\mathbf{x}})$ and the projection in the PCA space $P^l\varphi(\mathbf{x})$ is minimized, i.e.,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{M}} \|\varphi(\hat{\mathbf{x}}) - P^l\varphi(\mathbf{x})\|^2.$$

In terms of level-sets, this can be achieved by minimizing the following error

$$\begin{aligned}\rho(\hat{\psi}) &= \|\varphi(\hat{\psi}) - P^l\varphi(\psi)\|^2 \\ &= k(\hat{\psi},\hat{\psi}) - 2\langle\varphi(\hat{\psi}), P^l\varphi(\psi)\rangle + \|P^l\varphi(\psi)\|^2 \end{aligned} \tag{34}$$

where $P^l\varphi(\psi)$ is the projection of *any* SDF in kernel PCA space. Using the kernel notation, we can rewrite the middle term.

$$\begin{aligned}\langle\varphi(\hat{\psi}), P^l\varphi(\psi)\rangle &= \varphi(\hat{\psi})^T\Big[\sum_{n=1}^{l}\omega_n V_n + \bar{\varphi}\Big] \\ &= \varphi(\hat{\psi})^T\Big[\sum_{n=1}^{l}\omega_n\Big[\Big(\sum_{i=1}^{N}\frac{u_{ni}}{\sqrt{\lambda_n}}\ \varphi(\psi_i)\Big) \\ &\qquad\qquad - \bar{\varphi}\Big(\sum_{i=1}^{N}\frac{u_{ni}}{\sqrt{\lambda_n}}\ \Big)\Big] + \bar{\varphi}\Big] \\ &= \sum_{i=1}^{N}\tilde{\gamma}_i k(\hat{\psi},\psi_i)\end{aligned} \tag{35}$$

where

$$\gamma_i = \sum_{n=1}^{l}\frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} \qquad \text{and} \qquad \tilde{\gamma}_i = \gamma_i + \frac{1}{N}\Big(1 - \sum_{j=1}^{N}\gamma_j\Big).$$

Plugging the result of (35) into (34), the extremum can be obtained by setting $\nabla_{\hat{\psi}}\rho = 0$. Also, we prefer that the pre-image have a closed-form solution so that it can be used as basis for our energy functional in which we would like to evolve the KPCA shape weights directly in the input space.

4.1.3.1 Pre-Image for Linear PCA

We begin with a pre-image approximation for the polynomial kernel presented in Section 4.1.2.2. From this, we can then simplify the resulting pre-image approximation to that of linear PCA. By setting $\nabla_{\hat{\psi}}\rho = 0$, the following pre-image of the projection

in the KPCA space is

$$\begin{aligned}\hat{\psi} &= \sum_{i=1}^{N} \tilde{\gamma}_i \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right) \cdot \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right)^{-\frac{1}{d}} \psi_i \\ &= \sum_{i=1}^{N} \tilde{\gamma}_i \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right)^{\frac{d-1}{d}} \psi_i \\ &= \sum_{i=1}^{N} \tilde{\gamma}_i \left(\frac{c + \langle \hat{\psi}, \psi_i \rangle}{c + \langle \hat{\psi}, \psi \rangle} \right)^{d-1} \psi_i. \end{aligned} \tag{36}$$

However, if we use the approximation $\varphi(\hat{\psi}) \approx P^l \varphi(\psi)$, which amounts to assuming that $d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i)) \propto d^2_{\mathcal{H}}(\varphi(\hat{\psi}), \varphi(\psi_i))$, one has

$$k(\hat{\psi}, \psi_i) = \frac{1}{2}\big(\|P^l\varphi(\psi)\|^2 + k(\psi_i, \psi_i) - d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i)) \big)$$

Then, the following expression to reconstruct the pre-image is obtained:

$$\hat{\psi} = \sum_{i=1}^{N} \tilde{\gamma}_i \left(\frac{\|P^l\varphi(\psi)\|^2 + k(\psi_i, \psi_i) - d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))}{2\|P^l\varphi(\psi)\|^2} \right)^{\frac{d-1}{d}} \psi_i.$$

Interestingly, if we now take $d = 1$ in Equation (36), one gets the expression for performing linear PCA, i.e.,

$$\hat{\psi} = \sum_{i=1}^{N} \tilde{\gamma}_i \psi_i. \tag{37}$$

More importantly, we can now see that the pre-image approximation $\hat{\psi}$ presented in Equation (37) depends on the shape weight ω_i. In a similar manner, we can arrive at an energy formulation for nonlinear PCA. This is discussed next.

4.1.3.2 Pre-Image for NonLinear PCA

For the exponential kernel in Equation (33), which involves the implicit representation of SDFs, setting $\nabla_{\hat{\psi}} \rho = 0$ yields

$$\begin{aligned}\hat{\psi} &= \frac{\sum_{i=1}^{N} \tilde{\gamma}_i k(\hat{\psi}, \psi_i) \psi_i}{\sum_{i=1}^{N} \tilde{\gamma}_i k(\hat{\psi}, \psi_i)} \\ &= \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(- \|\hat{\psi} - \psi_i\|^2 / (2\sigma^2) \big) \psi_i}{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(- \|\hat{\psi} - \psi_i\|^2 / (2\sigma^2) \big)} \end{aligned} \tag{38}$$

While this expression has been used to estimate the pre-image via an iterative-based approach [54], we use the close-form approximation proposed by [71]. In their work, the authors assume that $\varphi(\hat{\psi}) \approx P^l\varphi(\psi)$. Moreover, they also assume that the Mercer kernel which is chosen can be inverted. Thus, for an exponential kernel of Equation (33), one has

$$d^2_{\mathcal{M}}(\hat{\psi}, \psi_i) = -2\sigma^2 \log\{\frac{1}{2}(2 - d^2_{\mathcal{H}}(\varphi(\hat{\psi}), \varphi(\psi_i)))\}.$$

Plugging this result into Equation (38) with the above approximation yields the following

$$\begin{aligned}\hat{\psi} &= \frac{\sum_{i=1}^N \tilde{\gamma}_i \exp\big(-d^2_{\mathcal{M}}(\hat{\psi}, \psi_i)/(2\sigma^2)\big)\psi_i}{\sum_{i=1}^N \tilde{\gamma}_i \exp\big(-d^2_{\mathcal{M}}(\hat{\psi}, \psi_i)/(2\sigma^2)\big)} \\ &\approx \frac{\sum_{i=1}^N \tilde{\gamma}_i \Big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\Big)\psi_i}{\sum_{i=1}^N \tilde{\gamma}_i \Big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\Big)}. \end{aligned} \tag{39}$$

Of course, the above approximations were made so that one can use kernel PCA to form a 3D shape embedded in a feature space $\mathcal{H}$ via the finite shape weights associated with the input space $\mathcal{M}$. However, before deriving the evolution scheme, let us first discuss the proposed pose estimation and segmentation framework.

4.2 Proposed Framework

We assume as with many machine learning techniques that we have a catalog of 3D shapes describing a particular object. Specifically, one can use stereo reconstruction methods [101] or range scanners to obtain accurate models as shown in Figure 20. From this, we derive a variational approach to perform the task of non-rigid 3D pose estimation and 2D image segmentation.

4.2.1 Some Notation and Terminology

Let S be the smooth surface in $\mathbb{R}^3$ defining the shape of the object of interest. With a slight abuse of notation, we denote by $\mathbf{X} = [X, Y, Z]^T$, the spatial coordinates that

are measured with respect to the referential of the imaging camera. The (outward) unit normal to S at each point $\mathbf{X} \in S$ will then be denoted as $\mathbf{N} = [N_1, N_2, N_3]^T$. Moreover, we assume a pinhole camera realization $\pi : \mathbb{R}^3 \mapsto \Omega; \mathbf{X} \mapsto \mathbf{x}$, where $\mathbf{x} = [x, y]^T = [X/Z,\ Y/Z]^T$, and $\Omega \in \mathbb{R}^2$ denotes the domain of the image I with the corresponding area element $d\Omega$. From this, we define $R = \pi(S)$ to be the region on which the surface S is projected. Similarly, we can form the complementary region and boundary or "silhouette" curve as $R^c = \Omega \setminus R$ and $\hat{c} = \partial R$, respectively. In other words, if we define the "occluding" curve C to be the intersection of the visible and non-visible region of S, then the image curve is $\hat{c} = \pi(C)$.

Now let $\mathbf{X}_0 \in \mathbb{R}^3$ and S_0 be the coordinates and surface that correspond to the 3D world, respectively. For example, if we choose the exponential kernel, S_0 is given as the zero-level surface of the following functional:

$$\hat{\psi}(\mathbf{X}_0, w) = \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \left(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l \varphi(\psi), \varphi(\psi_i))\right) \psi_i(\mathbf{X}_0)}{\sum_{i=1}^{N} \tilde{\gamma}_i \left(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l \varphi(\psi), \varphi(\psi_i))\right)}. \tag{40}$$

That is, $S_0 = \{\mathbf{X}_0 \in \mathbb{R}^3 : \hat{\psi}(\mathbf{X}_0, w) = 0\}$. Note, we have also kept the explicit dependency on w for ease of reading (e.g., when we compute the variation of this shape w.r.t. w in Section 4.2.3). Then one can locate the S in the camera referential via the transformation $g \in SE(3)$, such that $S = g(S_0)$. Writing this point-wise yields $\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$, where $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$.

4.2.2 Gradient Flow

Let us begin with the assumption that if the correct 3D pose and shape were given, then the projection of the "occluding curve," i.e. $\hat{c} = \pi(C)$, would delineate the boundary that optimally separates or segments a 2D object from its background. Further assuming that the image statistics between the 2D object and its background

are distinct, we define an energy functional of the following form:

$$E = \int_R r_o(I(\mathbf{x}), \hat{c}) d\Omega + \int_{R^c} r_b(I(\mathbf{x}), \hat{c}) d\Omega, \tag{41}$$

where $r_o : \chi,\ \Omega \mapsto \mathbb{R}$ and $r_b : \chi,\ \Omega \mapsto \mathbb{R}$ are functionals measuring the similarity of the image pixels with a statistical model over the regions R and R^c, respectively. Also, χ corresponds to the photometric variable of interest. In the present work, r_o and r_b are chosen to be region based functionals of [14, 66].

Now we want to optimize Equation (41) with respect to a finite parameter set denoted as $\xi = \{\xi_1, \xi_2, ..., \xi_m\}$. This is given as follows:

$$\begin{aligned} \frac{\partial E}{\partial \xi_i} = &\int_{\hat{c}} \Big(r_o(I(\mathbf{x})) - r_b(I(\mathbf{x})) \Big) \left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} \\ &+ \int_R \left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \xi_i} \right\rangle d\Omega + \int_{R_c} \left\langle \frac{\partial r_b}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \xi_i} \right\rangle d\Omega \end{aligned} \tag{42}$$

where the "silhouette" curve is parameterized by the arc length $\hat{s}$ with the corresponding outward normal $\hat{\mathbf{n}}$. Note, because we are only restricting r_o and r_b to be [14, 66], the last two terms can be shown to be zero. However, this is a special case, and one must take careful consideration when choosing the proper energy functional as these terms may tend not to be zero. We consider these energies for their simplicity, and more importantly, not to detract from the main contribution of the proposed method. In doing so, the result is as follows:

$$\frac{\partial E}{\partial \xi_i} = \int_{\hat{c}} \Big(r_o(I(\mathbf{x})) - r_b(I(\mathbf{x})) \Big) \left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s}, \tag{43}$$

If we further assume that the parameter ξ_i acts on the 3D coordinates, the above line integral will be difficult to compute since $\hat{c}$ and $\hat{\mathbf{n}}$ are defined in the 2D image plane. Thus, it would be much more convenient if we can express the above line integral around the "occluding curve" C that lives in the 3D space and is parameterized by s. We briefly describe this lifting procedure, and refer the reader to [24] for all the details. The image plane and surface are then related by

$$\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \xi_i}, J \frac{\partial \pi(C)}{\partial s} \right\rangle ds \tag{44}$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, which yields the following expression

$$\begin{aligned}
\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\
&= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds \\
&= \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds. \qquad (45)
\end{aligned}$$

Here K denotes the Gaussian curvature, and κ_X and κ_t denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$, respectively, where $\mathbf{t}$ is the vector tangent to the curve C at the point $\mathbf{X}$, i.e. $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$.

If we now plug the result of Equation (45) into Equation (43), we arrive at the following flow

$$\begin{aligned}
\frac{\partial E}{\partial \xi_i} = \int_C \Big(r_o\big(I(\pi(\mathbf{X}))\big) - r_b\big(I(\pi(\mathbf{X}))\big) \Big) \cdot \\
\frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds. \qquad (46)
\end{aligned}$$

Note, that in the above derivation we made no assumptions about the finite set. That is, we show that the overall framework is essentially "blind" to whether we optimize over the shape weights or pose parameters. What is important is how the functional in Equation 43 is lifted from the "silhouette" curve to the "occluding curve" so that the gradient can be readily computed. In particular, the term $\left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle$ is what we will focus on in Sections 4.2.3 and 4.2.4.

4.2.3 Evolving the Shape Parameters

In this section, we compute the term $\left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle$, when the ξ_i corresponds to the shape weight obtained from performing KPCA on a collection of 3D models. Let $\xi = \omega =$

$\{\omega_1, \omega_2, ..., \omega_l\}$ with l being the number of principal modes used. In addition, the 3D coordinates $\mathbf{X_0}$, which is derived from the surface S_0, are related by the constraint

$$\hat{\psi}(\mathbf{X}_0, w) = \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \left(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l \varphi(\psi), \varphi(\psi_i))\right) \psi_i(\mathbf{X}_0)}{\sum_{i=1}^{N} \tilde{\gamma}_i \left(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l \varphi(\psi), \varphi(\psi_i))\right)} \tag{47}$$

$$\text{s.t.} \qquad \hat{\psi}(X_0(w), w) = 0.$$

The term $\left\langle \frac{\partial \mathbf{X}}{\partial \omega_n}, \mathbf{N} \right\rangle$ can then be computed as follows:

$$\begin{aligned}
\left\langle \frac{\partial \mathbf{X}}{\partial \omega_n}, \mathbf{N} \right\rangle &= \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \omega_n}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N} \right\rangle \\
&= \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{R}^T \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{R}^T \mathbf{R} \mathbf{N}_0 \right\rangle \\
&= \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N}_0 \right\rangle.
\end{aligned} \tag{48}$$

Using the constraint on the zero-level surface and noting that $\frac{\nabla_{\mathbf{X}_0} \hat{\psi}}{\|\nabla_{\mathbf{X}_0} \hat{\psi}\|} = \mathbf{N}_0$, we then have that

$$\begin{aligned}
0 &= \frac{\partial}{\partial \omega_n} \hat{\psi}(\mathbf{X}_0(w), w) \\
&= \left\langle \nabla_{\mathbf{X}_0} \hat{\psi}, \frac{\partial \mathbf{X}_0}{\partial \omega_n} \right\rangle + \frac{\partial \hat{\psi}}{\partial \omega_n} \\
&= \left\langle \|\nabla_{\mathbf{X}_0} \hat{\psi}\| \mathbf{N}_0, \frac{\partial \mathbf{X}_0}{\partial \omega_n} \right\rangle + \frac{\partial \hat{\psi}}{\partial \omega_n},
\end{aligned}$$

which yields the following compact expression

$$\left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N}_0 \right\rangle = -\frac{1}{\|\nabla_{\mathbf{X}_0} \hat{\psi}\|} \cdot \frac{\partial \hat{\psi}}{\partial \omega_n}. \tag{49}$$

The general result presented in Equation (49) provides the variation of the energy with respect to the shape parameters, and is one of the major contributions of this work. It was previously shown that if one uses the linear PCA kernel, then $\frac{\partial \hat{\psi}}{\partial \omega_n} = V_n(\mathbf{X}_0)$. However, to exploit nonlinearities in the catalog of shapes, the exponential

kernel is employed. The variation of the pre-image for this kernel with respect to the shape weights is given as

$$\frac{\partial\hat{\psi}}{\partial\omega_i} = \frac{\sum_i^N \eta_i \cdot \psi_i}{\sum_i^N \mathcal{J}_i} - \frac{\big(\sum_i^N \eta_i\big)\big(\sum_i^N \mathcal{J}_i \cdot \psi_i\big)}{(\sum_i^N \mathcal{J}_i)^2}, \tag{50}$$

where

$$\mathcal{J}_i = \tilde{\gamma_i}\big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\big)$$

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\big) + ...$$
$$\frac{\tilde{\gamma_i}}{\sqrt{\lambda_n}}\big(\frac{1}{N^2}(\mathbf{l}^T\mathbf{K}\mathbf{l})\mathbf{l} - \frac{1}{N}\mathbf{K}\mathbf{l} + \mathbf{k}_{\varphi_\mathbf{i}} - \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\big)^T u_n - \omega_n.$$

and where $\mathbf{k}_{\varphi_\mathbf{i}} = [k_{\varphi_\sigma}(\psi_i, \psi_1), \ldots, k_{\varphi_\sigma}(\psi_i, \psi_N)]^T$.

For the complete derivation, we refer the reader to Appendix B. It is important to note though, that due to the closed form approximation of the pre-image, we are able to use various kernels (e.g., linear and nonlinear PCA) with minimal changes to overall scheme. Next, we discuss how one can evolve the pose parameters.

4.2.4 Evolving the Pose Parameters

In this section, we discuss the evolution of the pose parameters. Specifically, with a slight abuse of notation, we let $\xi = \lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}^T$. Then we are able to compute the term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$ where λ_i is a translation or rotation parameter:

- For $i = 1, 2, 3$ (i.e., λ_i is a translation parameter), and $\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$, one has

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle$$
$$= \left\langle \begin{bmatrix} \frac{\partial \lambda_1}{\partial \lambda_i} \\ \frac{\partial \lambda_2}{\partial \lambda_i} \\ \frac{\partial \lambda_3}{\partial \lambda_i} \end{bmatrix}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \delta_{3,i} \end{bmatrix}, \mathbf{N} \right\rangle \tag{51}$$
$$= N_i,$$

where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise).

- For $i = 4, 5, 6$ (i.e., λ_i is a rotation parameter), and using the expression of the rotation matrix written in exponential coordinates, $\mathbf{R} = \exp\left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix}\right)$, one has

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X_0}}{\partial \lambda_i}, \mathbf{N} \right\rangle$$
$$= \left\langle \mathbf{R} \begin{bmatrix} 0 & -\delta_{3,i} & \delta_{2,j} \\ \delta_{3,i} & 0 & -\delta_{1,i} \\ -\delta_{2,i} & \delta_{1,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle. \tag{52}$$

We note that one can also expand the rigid body transformation to a more general affine transformation, and hence provide increased flexibility in the proposed approach.

4.2.5 Alternative View of Gradient Flow

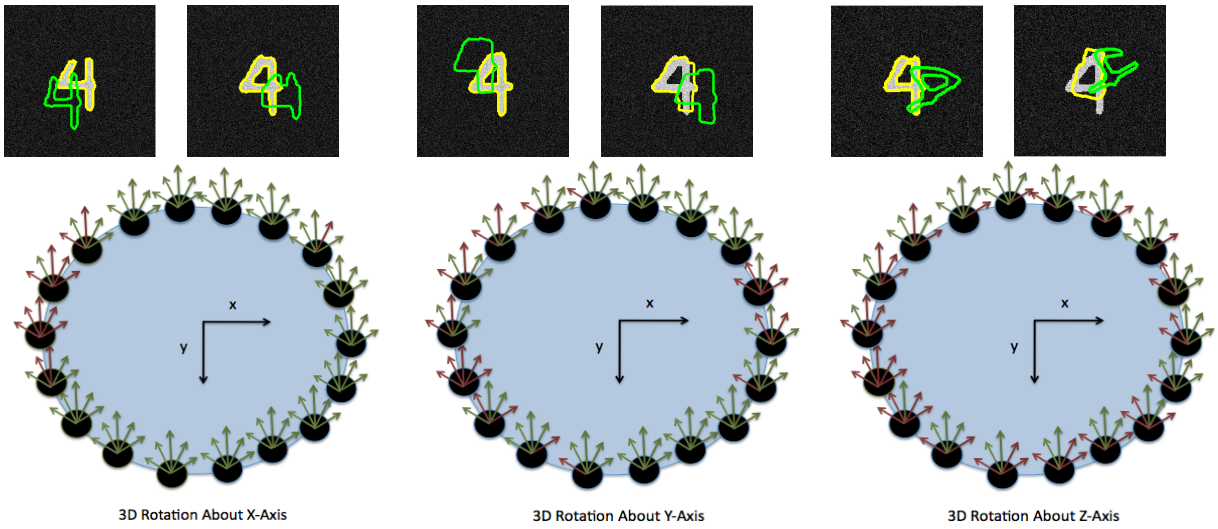
We briefly revisit the gradient flow of Section 4.2.2, and present an alternative viewpoint in relationship to the area of shape derivatives [15, 25].

In particular, revisiting equation (46), one can see that the gradient is the integral taken over the image curve. Alternatively, these points can be considered "voting" terms that bias the mean or direction of the particular finite parameter estimated.

Thus, one advantage as well as a limitation is the dependency on how well the statistical mean is for a given data set. Although it is shown in Section 4.4 that the proposed methodology exhibits an inherent robustness to occlusions in a tracking setting, if the occlusion obfuscates an object for extended period of time or possess a strong statistical difference from that of the intended object, then one might choose a different statistical measure. For example, naively choosing the median (during implementation) may allow for the algorithm to overcome certain occlusions during tracking. More importantly, one can also develop ad-hoc schemes or employ particle filtering techniques [85] to bias the original flow such that "outliers" on the image curve do not heavily affect the result. Unfortunately, given the scope of this work, experimental results presented in Section 4.4 are based upon the flow of Equation (46) and we leave the details of employing robust statistics as a subject of future work.

However, we note, that in certain scenarios in which there may be strong statistical difference between an occlusion and the object of interest for a extended period of time, some "occluded" points may drive the segmentation toward an unreliable result. Note, points that are only considered are those that lie on the image curve, and hence self-occluding points are taken care of during the computation of the occluding curve (see Section 4.3).

Moreover, the gradient in (43) involves the computation of the shape derivative, which describes the directions of deformation of the 2D curve (under projection) with respect to the 3D pose parameters and shape coefficients. The gradient is then the dot product of a typical 2D region-based gradient (i.e., $(r_o - r_b).\hat{\mathbf{n}}$; see e.g., Chan and Vese model [14]) with the shape derivative. We note that for each point on the 2D curve, the deformation direction is compared to the normal, $\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle$, and weights the statistical comparison term, $r_o - r_b$. Then the average over each point of the curve determines the optimal direction of variation of the finite parameter ξ_i (i.e., the



(a) (b) (c)

Figure 21: Domain of Convergence. a) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to x-axis. b) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to y-axis. c) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to z-axis. Note: Arrows are positioned at varying 30° increments and Red Arrows and Green Arrows denote "Failures" and "Success," respectively.

sign of the derivative $\frac{\partial E}{\partial \xi_i}$). Altogether, the link to shape optimization is presented here such that one can view the proposed methodology in the more broader text of shape based segmentation and pose estimation. Next, we discuss the numerical and implementations details associated with the algorithm.

4.3 Numerical Details

In Equation (46), the computation of the gradients involve the explicit determination of the occluding curve C. As previously mentioned, one can compute

$$C = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle = 0 \ \text{ and } \pi(\mathbf{X}) \in \hat{c}\}. \tag{53}$$

However, in practice, this condition is rarely exactly met due to the sampling of a 3D surface. Moreover, if the shape is non-convex, as is with most objects seen in this chapter, the condition of equation (53) will yield points that are self-occluded and hence truly not apart of the 3D occluding curve. Thus, an approximation of ϵ_1

is first made in order to compute an estimate of the visible and non visible regions, $\mathcal{V}^+_{\epsilon_1} = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle \geq -\epsilon_1\}$ and $\mathcal{V}^-_{\epsilon_1} = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle \leq \epsilon_1\}$, respectively. Relating the visible and non-visible regions to the occluding curve for both convex and non-convex shapes, we have $C^+ = \{\mathbf{X} \in \mathcal{V}^+_{\epsilon_1} : \exists\, \mathbf{Y} \in \mathcal{V}^-_{\epsilon_1}, \|\mathbf{X} - \mathbf{Y}\| \leq \epsilon_2\}$ and $C^- = \{\mathbf{X} \in \mathcal{V}^-_{\epsilon_1} : \exists\, \mathbf{Y} \in \mathcal{V}^+_{\epsilon_1}, \|\mathbf{X} - \mathbf{Y}\| \leq \epsilon_2\}$ where ϵ_2 is a chosen (small) parameter. The occluding curve can then be redefined as the union of these two sets or $C = \{\mathbf{X} \in C^+ \cup C^-\}$. We have seen that this procedure mitigates self-occluded points if the proper choice of ϵ_1 and ϵ_2 are chosen with regards to the sampling of the 3D surface and camera calibration parameters.. In addition, $\hat{c}$ can be obtained by using morphological operations on $R, i.e., \hat{c} = R - \mathcal{E}(R)$, with $\mathcal{E}$ denoting the erosion operation for a chosen kernel [38].

Secondly, to save computational time, we approximated the term $\sqrt{\frac{\kappa_X \kappa_t}{K}} \simeq 1$ in Equation (46). We note that the approximation was done in order to save computational time and indeed can be a poor approximation if the viewing direction $\mathbf{X}$ and the tangent to the occluding curve are identical. However, we observed that for the sequences and images presented in this chapter, the energy decreased and convergence was met.

Lastly, in performing KPCA, and as with any other statistical learning technique, one must choose the number of modes of variation. In this chapter, depending on the data set, the number of modes was chosen to be $l = 4, 5$ or 6. Moreover, when working with the exponential kernel, the choice of σ is important. Hence, we found that if we choose $\sigma^2 = \frac{1}{N} \sum_i \min_{j, i \neq j}(\|\psi_i - \psi_j\|^2)$, the algorithm would, for our particular experiments, converge to the desired shape of interest. If one would like to "mix" the shapes in a more linear fashion, σ should be chosen to be a higher value and vice versa.

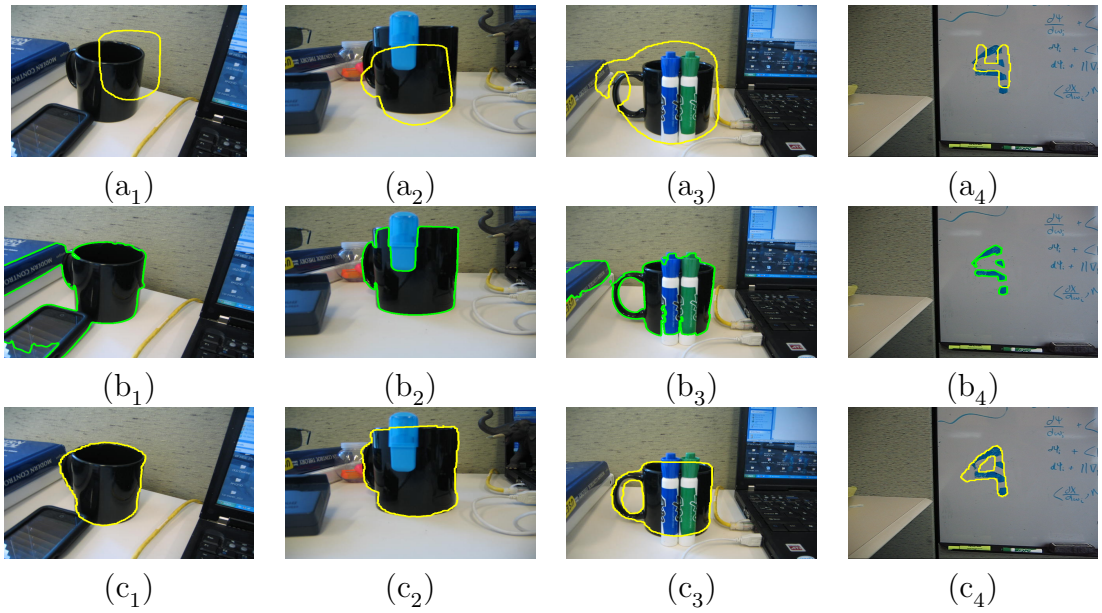
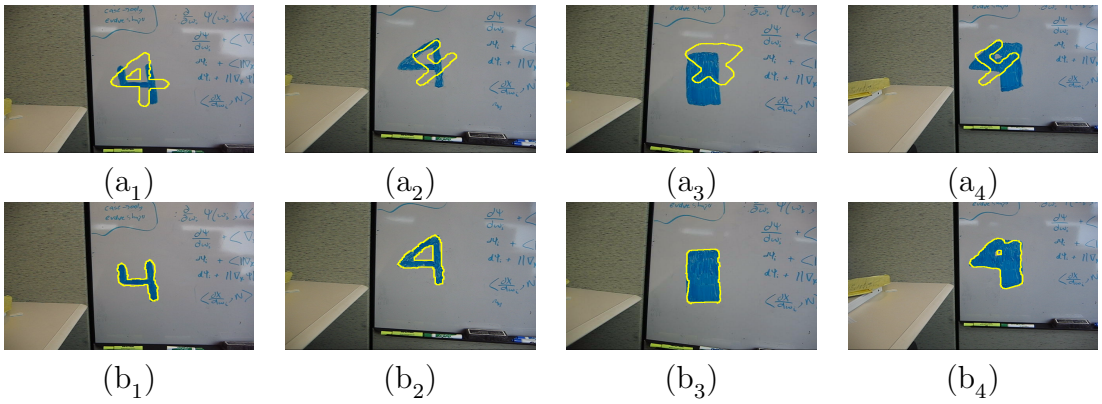


Figure 22: Linear PCA Segmentation with Occlusion and Clutter. *Top Row:* Initialization *Middle Row:* Unsatisfactory results obtained from using an active contour. *Bottom Row:* Results obtained from proposed approach

4.4 Experiments

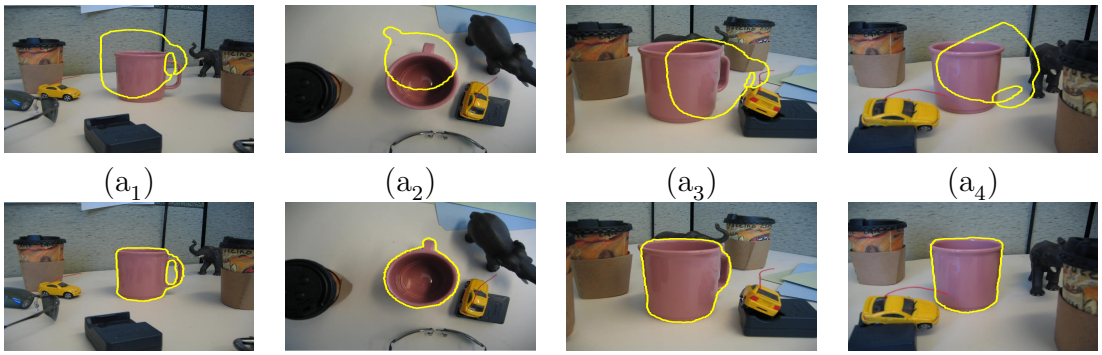
We provide segmentation and tracking results to demonstrate the algorithm's **domain of convergence** and its ability to handle **noise**, **deformation**, **occlusions** or **clutter**. Moreover, results are given to illustrate the method's effectiveness in **shape recovery**, which may include **nonlinearities** in one's training set. Specifically, we generate three 3D training sets corresponding to the number "4" as well as commonly seen tea cups and helicopters [2] . This is shown in Figure 20. Lastly, because code was not readily available, it should be noted that we do not claim the proposed method is superior (practically) to existing techniques. Thus, the experiments were performed to highlight the (dis)advantages of a proposed alternative approach for non-rigid segmentation and pose estimation.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

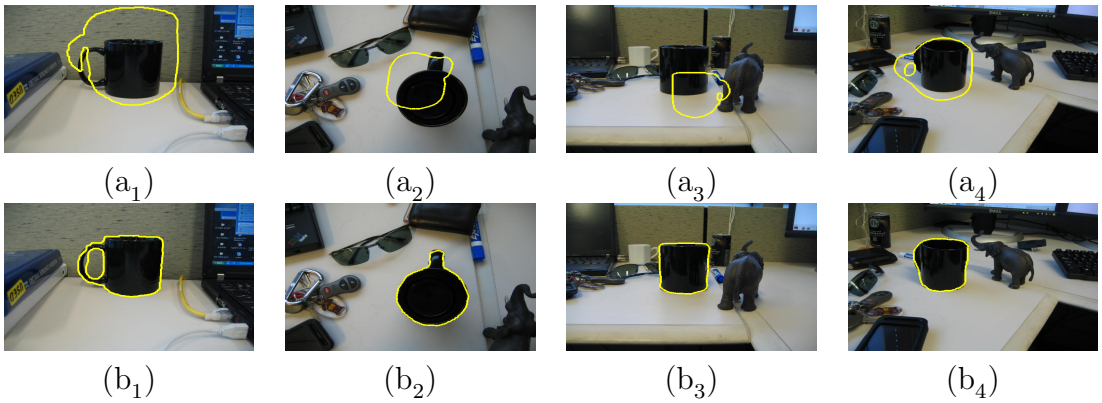
Figure 23: Linear PCA Segmentation for Shape Recovery: Segmentation of different views and shapes of the number "4," which are not present in the training set. *Top Row*: Initialization. *Bottom Row*: Final Results obtained for running proposed.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

Figure 24: Linear PCA Segmentation for Shape Recovery: Segmentation of different views and shapes of a beige teacup, which is not present in the training set *Top Row*: Initialization. *Bottom Row*: Final Results obtained for running proposed.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

Figure 25: Linear PCA Segmentation for Shape Recovery: Segmentation of different views and shapes of a black teacup, which is not present in the training set *Top Row*: Initialization. *Bottom Row*: Final Results obtained for running proposed.

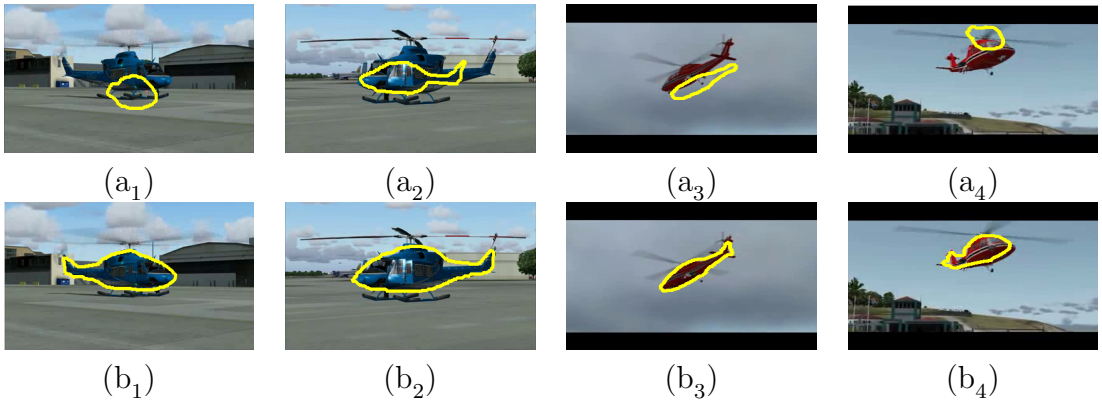


Figure 26: Nonlinear PCA Segmentation for Shape Recovery: Segmentation of different views and shapes of simulated helicopters which are not present in the training set. *Top Row*: Initialization. *Bottom Row*: Final Results obtained for running proposed.

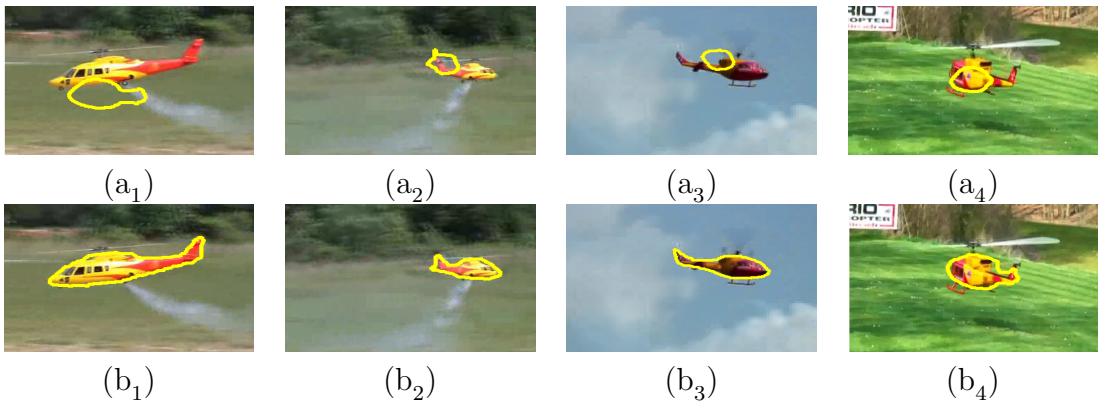


Figure 27: Nonlinear PCA Segmentation for Shape Recovery: Segmentation of different views and shapes of real helicopters which are not present in the training set. *Top Row*: Initialization. *Bottom Row*: Final Results obtained for running proposed.

4.4.1 Domain of Convergence

In the first set of experiments, we illustrate the proposed method's domain of convergence. Although the algorithm is designed for a variational setting such as tracking in which movements between frames are assumed to be small, it is still interesting to note the domain of convergence.

Specifically, we generated 18 random pairs of the synthetic number 4. This was

[2] After the models were generated, the shapes were registered with the methodology proposed in [85] so that any variation in shape was not due to pose alignment error

done by first setting the number of modes $l = 4$, and randomly drawing the set of nonlinear PCA weights from a uniform distribution $\mathcal{U}(2\sigma_i^{\text{deg}}, 2\sigma_i^{\text{deg}})$ where σ_i^{deg} is i^{th} primary mode of variance. Figure 21 shows several sample shapes that were generated as well their initializations and final result shown in green and yellow, respectively. In particular, using the validation manner to that of [80, 91], one pair was fixed at the center of the image while the other shape of a specific pair was initialized at positions sampled on a circle whose radius is approximately half of the fixed shape width. Note, the depth was kept constant when initializing the "moving" shape. Then initial rotations of $\pm 60°$ in $30°$ increments were tested about each rotational axis while keeping the other two remaining axis fixed. The bottom row of Figure 21 shows the sampled circle for each of the rotations and through user visualization, each arrow was marked red for which we considered a "failure" while green represented a "successful" segmentation.

Interestingly, we see that for rotations about the x and y axis, the algorithm exhibits a larger convergence region then for the z axis. That is, of the 90 possible initializations, we successfully segmented 76 when rotation was applied to the x axis, and successfully segmented 66 when rotation was applied to the y axis. On the other hand, only 48 successful segmentations were observed when rotating about the z axis. The resulting behavior can be mostly attributed to the initial overlap the "moving" shape was given. That is, for specific rotations, a change in perspective is seen in the 2D image domain. Combining this with large translations and deviations from the shape result in initializations seen in Figure 21, whereby the algorithm is driven to an undesirable result. However, we should note that this a drawback associated with gradient descent as opposed to another optimization method. Although it is beyond the scope of this work, one could strap a particle filter in order to further widen the domain of convergence at the cost of computational complexity. Nevertheless, the domain of convergence shown here is ideal for segmentation in which one knows

the approximate location of the object or tracking scenarios in which deviations in location are generally not seen between consecutive frames. Lastly, we also make note that we performed this test with the linear PCA kernel described earlier and similar results were obtained.

4.4.2 Segmentation Experiments

In this section, we focus on segmenting a 2D image using a 3D catalog of shapes that may or may not correspond to the object of interest.

4.4.2.1 Linear PCA Segmentation with Occlusions and Clutter

For the second set of experiments, we provide experimental validation that compares our segmentation method, which is an optimization over a finite set of parameters, with that of the infinite dimensional geometric active contour (GAC) technique. The reasoning for such a comparison is as follows: In either methodology, we seek to minimize a cost functional of the general form $E(t) = \int_{\gamma} \Psi(\mathbf{x}, t) d\mathbf{x}$ over a family of curves. For the GAC framework, this family of curves lives only in the image plane when performing 2D image segmentation. Similarly, in the current framework, we are optimizing over a family of 3D "occluding" curves that correspond to 2D "silhouette" curves. That is, we cast the typical infinite dimensional problem of segmentation as a finite dimensional optimization problem.

Thus, we benefit from incorporating shape information which results in being able to deal with not only occlusion, but also in cluttered environments where the original assumption of separable statistics does not hold. This is shown on four different examples as seen in Figure 22. The top row illustrates the initialization, while the middle row shows the unsatisfactory result of using an active contour. The final row highlights results given by the proposed approach with $l = 6$ using the energy proposed in [14]. Although it is not readily apparent, one can alternatively view the examples in Figure 22 as 3D reconstruction from a single 2D image view exhibiting

partial information, which is a fundamental task in computer vision. Details on the level of accuracy of average reconstruction can be found in Section 4.4.2.4 and Section 4.4.3.1. Also, we note that these experiments can be found in [85]

4.4.2.2 Linear PCA Segmentation for Shape Recovery

In this section, we shift the focus of our experiments to segmenting shapes that are not in the original training sets using the linear KPCA kernel as done in [85]. We want to specifically highlight an important advantage of learning 3D shapes as opposed to a large catalog of 2D shapes to perform the task of 2D segmentation. This is done by segmenting different shapes arising from different 3D models as well as from altering the pose of a 3D object.

In Figure 23, we show the segmentation of different views and shapes that are obtained from a person coloring in the number "4" on a white-board. The top row highlights the initialization while the bottom rows shows the final result. The same experiment is performed for the differing views of two tea cups that were not in the original training set (see Figure 24 and Figure 25). While one may argue that the results are similar to the 2D shape learning approaches, we should note some of the key differences. First, we not only segment the 2D image, but are able to return the estimated 3D shape and pose from which the 2D object was derived. Moreover, to account for segmentation of objects presented in Figure 23, Figure 24, and Figure 25 with a 2D shape prior, one would have to learn every possible projection of the 3D object onto to the 2D image plane (if no prior knowledge is given about the aspect of the projection). Note, we set $l = 6$ and used the energy proposed in [66]

4.4.2.3 Nonlinear PCA Segmentation for Shape Recovery

Although it was shown in the previous section that one can segment an object from various views with linear PCA, we now shift our focus on utilizing the nonlinear PCA

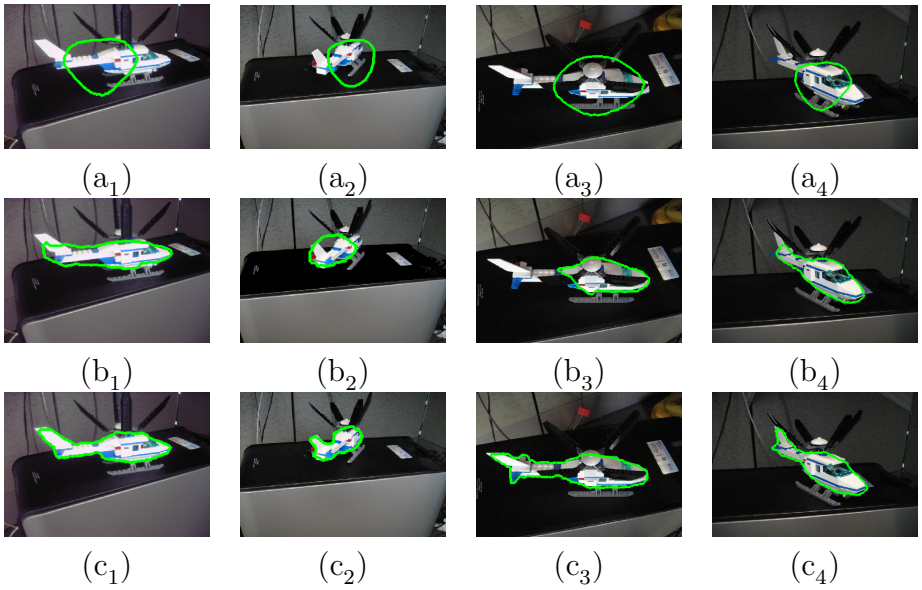


Figure 28: Nonlinear vs Linear PCA Segmentation: Image segmentation results when two different class of shapes (teacups and helicopters) are "mixed" to generate a new training set. *Top Row:* Initialization. *Middle Row:* Unsatisfactory results obtained using proposed algorithm with PCA. *Bottom Row:* Satisfactory results obtained using proposed algorithm with KPCA.

kernel for a more complex training set such as a rigid class of helicopters. In particular, we found several images and differing viewpoints of helicopters available online (http://www.youtube.com). Thus, the objects again were not within the training set. In particular, Figure 26 focuses on segmenting a Sikorsky S76 as well as a Bell 212 simulated helicopter from two different views. We should note that while these objects are simulated, the scene is also cluttered with simulated real-life clutter. Despite relatively difficult initializations, we were able to successfully segment two distinct helicopter used. On the other hand, Figure 27 shows successful segmentations of the same type of helicopters, but in a real-time environment. Note, we set $l = 5$ and used the energy proposed by [66] for these set of experiments

4.4.2.4 Nonlinear vs Linear PCA Segmentation

The next set of experiments demonstrate the robustness of utilizing a nonlinear kernel as opposed to linear PCA in the context of the proposed algorithm. A major

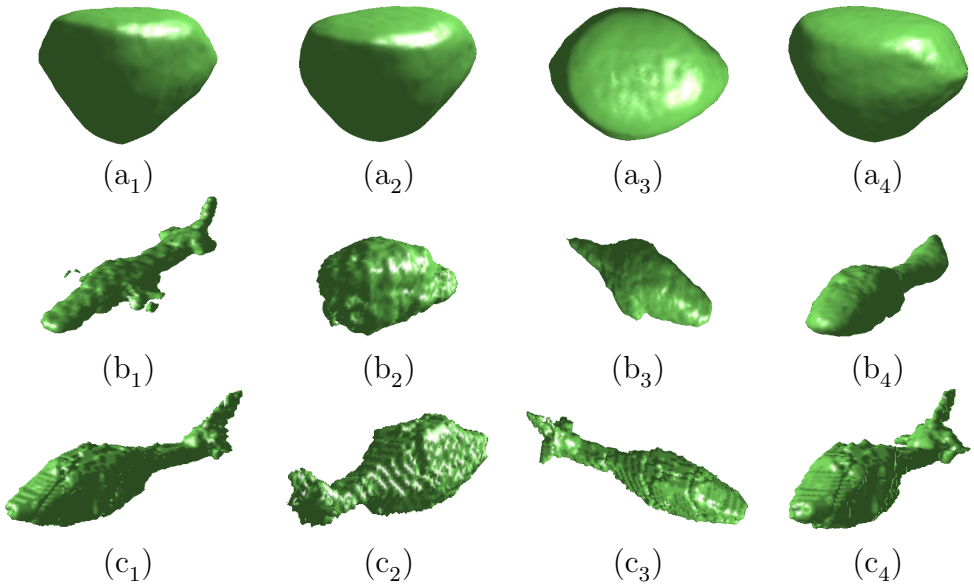


Figure 29: Nonlinear vs Linear PCA Segmentation: 3D shape reconstruction results when two different class of shapes (teacups and helicopters) are "mixed" to generate a new training set. *Top Row:* Initialization. *Middle Row:* Unsatisfactory results obtained using proposed algorithm with PCA. *Bottom Row:* Satisfactory results obtained using proposed algorithm with KPCA.

drawback in using linear PCA is that one assumes the training set can be linearly related and decomposed to form a novel learnt shape. This assumption is many times invalid. For example, when dealing with a more general catalog of shapes, in which two sets of objects from different classes are mixed, linear PCA will likely yield an unsatisfactory result. Thus, in order to deal with this problem, one may employ (again) a nonlinear statistical learning technique such as nonlinear PCA. We note that this problem is widely known in literature and has been solved for various 2D shape based schemes [61, 21]. However, for the sake of completeness, we demonstrate the increased performance of using nonlinear PCA as compared to linear PCA in the present framework.

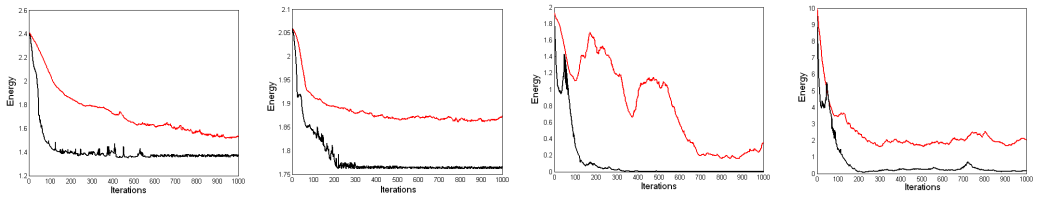
We begin by mixing two of the 3D training sets together. In particular, we "corrupt" the helicopter training set, which consists of 12 models, by adding 8 of the commonly used teacups. This "new" set of training models is used to perform 2D segmentation of a toy helicopter, which is not in our training set, for several different

viewing angles as seen in Figure 28. The first row highlights the initialization while the second and third row yield the results of the proposed algorithm when employing linear PCA and nonlinear PCA, respectively. Moreover, because our algorithm can be alternatively viewed as a reconstruction methodology from a single 2D image [79], we also present the corresponding initialization and results for 3D shapes in Figure 29.

Interestingly, while it may be apparent that nonlinear PCA outperforms linear PCA from the perspective of shape learning, we also found, on average, a faster convergence rate for nonlinear PCA from both a 2D segmentation point of view as well as for shape reconstruction. This "distance" is computed via ICP measure between two sampled set of points that lie on the surface of each respective 3D shape [5]. This is shown in Figure 30 for two of the four viewing angles of our toy helicopter. Specifically, the top row shows the normalized 2D image segmentation energy of [14], while the bottom row shows the normalized L_2 distance between the known 3D shape with that of its reconstructed 3D shape. One can see from the above plots that although the 2D image energy is always being minimized until it reaches steady-state, the shape error may gradually increase before it decreases. This is because we are optimizing with the image segmentation energy, and not with the L_2 distance associated with the shape reconstruction error. Nevertheless, nonlinear PCA exhibits a faster convergence rate for both shape reconstruction and image segmentation. Note, we set $l = 5$ for both linear and nonlinear PCA kernels.

4.4.3 Tracking Experiments

In this section, we focus on tracking an object in a 2D scene using a 3D catalog of shapes. In particular, our initialization for each frame in the video sequence is the pose and shape result from the previous frame. That is, it can be viewed as an extension of the segmentation algorithm discussed previously.



(a) (b) (c) (d)

Figure 30: Nonlinear vs Linear PCA Segmentation: PCA and KPCA comparison convergence plots of image and shape energy. (a)-(b) Segmentation energy convergence plot of two different examples of segmenting a toy helicopter. (c)-(d) Corresponding shape energy convergence plots. Note: Black color denotes KPCA result while red color denotes PCA result.

4.4.3.1 Linear PCA Tracking of Synthetic Deformations, Occlusions, and Noise using the Number 4

In the first set of tracking experiments, we demonstrate the algorithm's robustness to noise in the presence of continuous deformation and occlusion.[3] First, a tracking sequence was generated consisting of 200 frames that were obtained from projecting the number "4" onto the 2D image plane using a simulated camera. Specifically, the variation in the rotation angle was a complete 360° cycle, and the model was varied linearly along its z-axis from 170 to 670 spatial units resulting in a scale in the viewing aspect of the image projection. Also, we translated the model in its x-y axis with a step of 0.5 spatial units for each frame. More importantly, we vary the first three principal modes so that a deformation can be seen. From this basic sequence, three cases of additive Gaussian noise with a standard deviation of $\sigma_n = 25\%$, $\sigma_n = 75\%$, and $\sigma_n = 100\%$ are formed. We also generated an artificial occlusion resembling the intensity of the background with noise $\sigma_n = 25\%$. [4]

Figure 31 show four frames which exhibit typical tracking results from the $\sigma_n = 75\%$ noise case described as well as the generated occlusion sequence. Here, we

[3]We note that a similar, but not exact, experiment can be found in [85]. Also, performing this experiment with nonlinear PCA would yield similar results due to the nature of the given data set.

[4]Note: we define σ_n to be a percentage of noise generated from a gaussian distribution to be applied to its corresponding binary image. For example, given the number 4 and the case of $\sigma_n = 25\%$, the image value of $\mathcal{N}(1, .25)$ and $\mathcal{N}(0, .25)$ is chosen for the object and background, respectively.

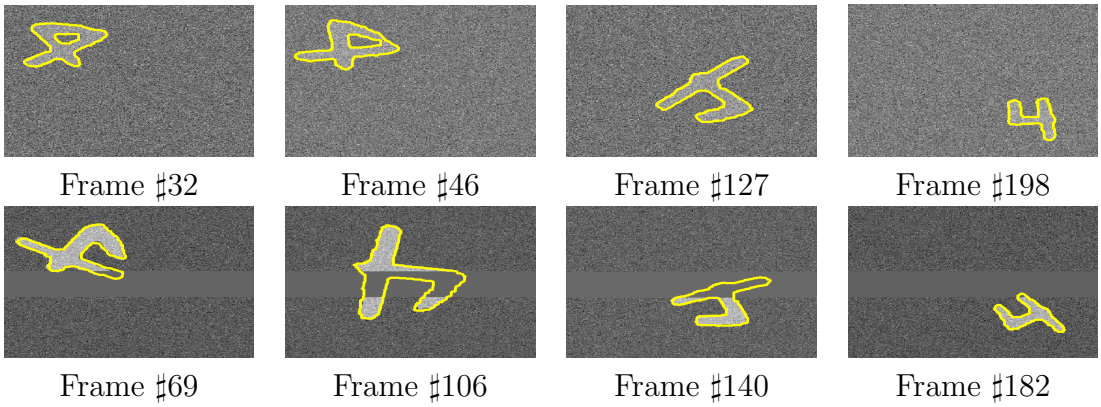


Figure 31: Linear PCA Tracking of Synthetic Deformations, Occlusions, and Noise using the Number 4. Visual tracking results for the sequence involving the number 4. *Top Row:* Tracked sequence with Gaussian noise of standard deviation $\sigma = 75\%$. *Bottom Row:* Tracked sequence for $\sigma = 25\%$ with severe occlusion.

have used the region-based energy of [14], and obtain tracking results by varying the first 6 principal modes, i.e. $l = 6$. Because several 2D-3D pose estimate techniques [70, 28, 59] rely on a correspondence based scheme to estimate the pose, their methodologies may be sensitive to noise and outliers as presented in Figure 31. Because of the robustness of region-based active contours (as compared to local geometric descriptors), the proposed approach yields satisfying visual results such as those of [24]. However, here we have not constrained ourselves to know the actual shape of a pre-specified number "4". It is straightforward to see (without comparison), that if we were to assume the knowledge of a model, then it would not be possible for us to handle the wide range of deformations seen.

Nevertheless, to further ensure the robustness of the algorithm, we provide quantitative tracking results as seen in Table 5. For each image, percent *absolute* errors with respect to the ground-truth were computed for both the translation and rotation as: Error $= \frac{\|\mathbf{v}_{\text{measured}} - \mathbf{v}_{\text{truth}}\|}{\|\mathbf{v}_{\text{truth}}\|}$, with $\mathbf{v}$ a translation or quaternion vector. In dealing with the appropriate shape error, we opted to compute the error as: Error $= \frac{\|\mathbf{X}_{\text{measured}} - \mathbf{X}_{\text{truth}}\|}{\max_{\text{j,i}\neq\text{j}}(\|\mathbf{X}_{\text{i}} - \mathbf{X}_{\text{j}}\|)}$. Note, $\mathbf{X}$ represents the 3D shape of interest, and overload of notation is employed only for this section. Also i and j are simply indices belonging to a training set.

Noise Level	**mean error.** (in %)	**std. dev. error** (in %)	**max error** (in %)
25%	**T:** 1.39 **R:** 1.08 ω : 3.46	**T:** .44 **R:** 0.36 ω : 0.84	**T:** 2.63 **R:** 2.15 ω : 5.87
75%	**T:** 1.67 **R:** 1.46 ω : 4.20	**T:** .61 **R:** 0.48 ω : 0.88	**T:** 3.41 **R:** 2.95 ω : 6.65
100%	**T:** 1.85 **R:** 1.41 ω : 4.24	**T:** 0.69 **R:** 0.39 ω : 0.97	**T:** 3.62 **R:** 2.14 ω : 7.02
25% (Occ.)	**T:** 2.51 **R:** 2.75 ω : 5.27	**T:** 0.85 **R:** 1.38 ω : 0.91	**T:** 4.94 **R:** 6.93 ω : 7.89

Table 5: Quantitative tracking results demonstrating robustness to deformation (and noise). Mean, Standard Deviation, and Max Errors with respect to translation, rotation, and shape recovery are reported for several levels of noise.

At any rate, this error allows for us to see how accurate our shape reconstruction is with regards to the maximal error seen in one's training set. From Table 5, we see that the errors for rotation and translation are small even in the presence of noise or severe occlusion. On the other hand, we do see a slight increase in shape error with respect to rotation and translation. This can be mainly attributed to the fact that pose changes may account for varying shapes as well as the fact that we are only dealing with a single 2D image. However, it still remains small and visually correct.

Moreover, if it is desirable to track a rigid object that is representative of a certain class (e.g., cars, boats, or planes), then one can learn the different 3D models of a class, and thereby relax the constraint of the prior knowledge needed. This will be demonstrated next.

Experiment	2D Image Size	3D Model Size	Number of Modes Used	Number of of Iterations	Time (in Sec)
Linear PCA Occlusion and Clutter (Black Teacup)	[640X480X3]	[128X128X128]	6	1000	18.14
Linear PCA Shape Recovery (Number 4)	[640X480X3]	[100X100X50]	6	500	7.31
Linear PCA Shape Recovery (Beige Teacup)	[640X480X3]	[128X128X128]	6	1000	18.29
Nonlinear Training Set (Toy Helicopter)	[640X480X3]	[128X128X128]	4	1200	22.64
Tracking Deformation and Noise (Number 4)	[300X300X1] X 200	[100X100X50]	6	30/Frame	155
Tracking Occlusion and Clutter (Toy Helicopter)	[512X288X3] X 1319	[128X128X128]	5	40/Frame	1022

Table 6: Performance Analysis of Proposed Non-Rigid 2D3D Pose Estimation and Segmentation Algorithm. Note image sizes of [N X K] and [N X K X 3] represent grayscale and color images.

4.4.3.2 Nonlinear PCA Tracking Occlusion and Clutter of a Toy Helicopter

In many rigid tracking scenarios, only the general class of the object of interest may be known and given. In this experiment, we track a sequence involving a toy helicopter that is not in our 3D helicopter training set. In particular, the sequence presents not only aspect and view changes, but also occlusions from the helicopter rotor blades and a human hand guiding the object. These occlusions are particularly difficult from a statistical viewpoint. That is, the rotor blades are visually black as opposed to a mostly white helicopter, which may cause many segmentation algorithms to exclude this portion of the object. In contrast, the light appearance of the hand relative to most of the background may result in leaks or capture the hand entirely along with the helicopter. Moreover, the clutter in the scene may cause additional problems as previously discussed in Section 4.4.2.1.

Using the region-based energy of [14], we are able to obtain tracking results by varying the first 5 principal modes, i.e. $l = 5$. Figure 32 show several frames of the video sequence. In particular, the sequence begins with the helicopter on the ground with its rotor blades moving. This inherent movement of the object causes self-occlusions. As the blades begin to slow, the helicopter is moved and placed on top of book. We see occlusions from not only the blade, but also the human hand. Nevertheless, the algorithm successfully maintains track. However, we do note that improved tracking performance could be gained by taking into account the temporal coherency of the sequence and using filtering principles such as those proposed in [8, 73].

4.4.3.3 Nonlinear PCA Tracking of a Sikorsky S-76 and Bell 212 Helicopter in Real Scenarios

Extending the previous section, we now track two helicopters from videos that were obtained via online (http://www.youtube.com). In particular, using the region-based energy of [66], we were able to obtain tracking results by varying the first 5 principal

modes, i.e, $l = 5$ for both the Sikorsky S76 and Bell 212 sequence as seen in Figure 33 and Figure 34, respectively.

In the sequence shown in Figure 33, the Sikorsky S76 sequence exhibits a large change in the rotation caused by both the helicopter as well the camera itself. This resulted in several aspect changes like that of Section 4.4.3.2. However, the scene in which the helicopter flies in, while simulated, is realistic given the amount of clutter and buildings surrounding the helicopter. Nevertheless, tracking is maintained throughout the sequence. In Figure 34, we track a Bell 212 helicopter in a real life sequence. We should note here that although track is effectively maintained, it can be seen the segmentation is not as successful as that of Figure 33. That is, one can see that the contour does not properly segment the tail end of the helicopter. This phenomenon can be attributed to the density of the training set. Although we have mentioned that we are able to reduce the computational complexity with regards to 2D shape based learning, the algorithm can suffer the classic drawbacks when specific details of an object are left out (e.g., a specialized rotor). At any rate, track is still maintained despite these difficulties associated with shape based learning techniques.

4.4.4 Performance Analysis

In this section, we report the execution time, iteration count, and other relevant information for several of the experiments performed in this chapter in Table 6. It should be noted that our implementation of the proposed algorithm was done in both MATLAB v7.1 as well as C/C++ on a Intel Dual Core 2.66GHz with 4 GB memory. The two pieces of code were integrated via MEX. Also, in relative terms, our implementation of running just the pose portion of the algorithm averaged nearly 100 iterations per second. When including and optimizing over the shape weights, the main reduction in speed was the volume size of the 3D shape as well as the number of models used in the given training set. The 2D image size also impacts computational

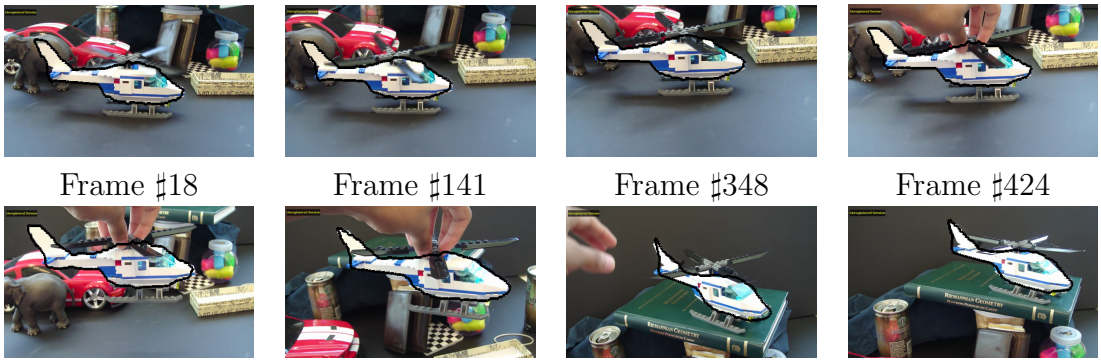


Figure 32: Tracking with Occlusion and Clutter (Toy Helicopter). Several tracking frames are shown.

speed, but is not significant. Moreover, the algorithm can be seen as "unoptimized" code since several steps can be done to effectively increase computing in a sparse manner (e.g., keeping a list of occluding curve points). However, this is beyond the scope of this thesis, and future work is planned to address this issue in detail.

4.5 Chapter Conclusion

In this chapter, we derive a geometric and variational approach to perform the task of 2D-3D *non-rigid* pose estimation and 2D image segmentation. This can be seen as a extension of the framework presented in [24, 85], where we assume that we have only a single 3D shape prior. Instead, in the present work, we infer a 3D shape prior from a catalog of 3D rigid shapes, which may represent a general class of an object or possibly a set of deformations that may occur to the model. In addition, to account for nonlinearities in the given training set, we showed that other statistical learning techniques can be employed with minimal changes to the overall framework. As a result, we fully exploit the task of pose estimation and segmentation in a unified framework.

Moreover, because we have shown the above framework is ideal for tracking rigid objects of a certain class, a direction for future work would be to employ filtering principles such as particle filtering [73, 80]. This is of particular importance since

Figure 33: Tracking a Sikorsky S-76 through a Simulated Environment. Several tracking frames are shown.

Figure 34: Tracking a Bell 212 through a Real Environment. Several tracking frames are shown.

it has been shown that if one exploits the inherent temporal component in video sequences, even more robust tracking results can be obtained. We believe that this should improve the algorithm's ability to deal with even more challenging occlusions and environments.

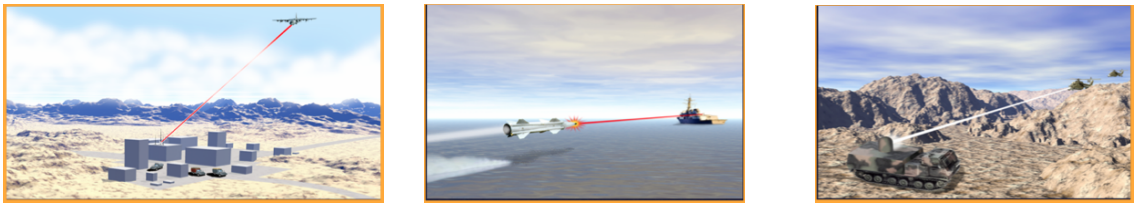
CHAPTER V

APPLICATIONS: TACTICAL TRACKING WITH 3DLADAR IMAGERY

A well studied problem in controlled active vision is the fundamental task of tracking moving and deformable objects [8, 93]. In this chapter, we present a novel tracking scheme using imagery taken with the 3-D Laser Radar (3DLADAR) system developed at MIT-Lincoln Labs [2]. Specifically, the system employs the Active Contour Framework [13, 52, 14], the algorithm proposed in Chapter 3 as well as a detection algorithm based on appearance models to efficiently track a "target" under challenging scenarios. As opposed to typical 2D systems [98, 6], in which tracking is limited by a lack of contrast, 3DLADAR provides the capability to improve aim point tracking by encoding range and angle-angle information as seen in Figure 35. We can tackle the problem with the following two pronged appraoch:

- First, use image segmentation in a coarse manner to capture 'target" features.
- Second, use a point set registration scheme to align features to template model.

It is here that we should note one can and perhaps should employ the algorithm developed in Chapter 4 to solve this tracking problem. However, due to the fact that we are dealing with range imagery, the equivalent problem is to capture "enough" target features so that those features can be aligned to a template model and the 3D pose is retrieved. That is, we are not concerned with segmenting the entire target as in Chapter 4, but rather employ segmentation in a coarse manner in which we favor under segmented results (i.e., we want to have complete faith that the result should not contain background information). This is turn will limit the number of "outliers"



(a) Air-Ground (b) Horizontal (c) Ground-Air

Figure 35: Several Environments for which 3DLADAR is employed.

in point set registration. Nevertheless, if range imagery is not provided, we advocate the use of the method in Chapter 4.

5.1 Preliminaries

In this section, we introduce two new concepts for this thesis: Segmentation with Thresholding Active Contours (TAC) and Appearance Models. These concepts along with the method presented in Chapter 3 will be essential for the proposed tracking algorithm.

5.1.1 Thresholding Active Contours (TAC)

To define the TAC, consider an image I over the domain $\Omega \in \mathbb{R}^2$ that is partitioned into regions by an evolving contour C, where C is embedded as the zero level set of a signed distance function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $C = \{x|\phi(x) = 0\}$ [88, 64]. The shape of the evolving contour is described by the Heaviside function, $H\phi$, which is 1 when $\phi < -\epsilon$, 0 when $\phi > \epsilon$, and has a smooth transition in the interval $[-\epsilon, \epsilon]$. Similarly, the interface at the zero level set can be denoted by $\delta\phi$, the derivative of $H\phi$, which is 1 when $\phi = 0$ and 0 at distance ϵ from the interface.

To incorporate statistical information into the segmentation, let us denote the probability of a point being located inside or outside of the curve as G_{in} or G_{out} respectively. Furthermore, we assume that these probabilities are Gaussian: $G_{\text{in}} = \mathcal{N}(\mu_{\text{in}}, \Sigma_{\text{in}})$ and $G_{\text{out}} = \mathcal{N}(\mu_{\text{out}}, \Sigma_{\text{out}})$, where $\mathcal{N}$ denotes the normal distribution. In the case of 3DLADAR, reflectance and range data represent two linearly independent

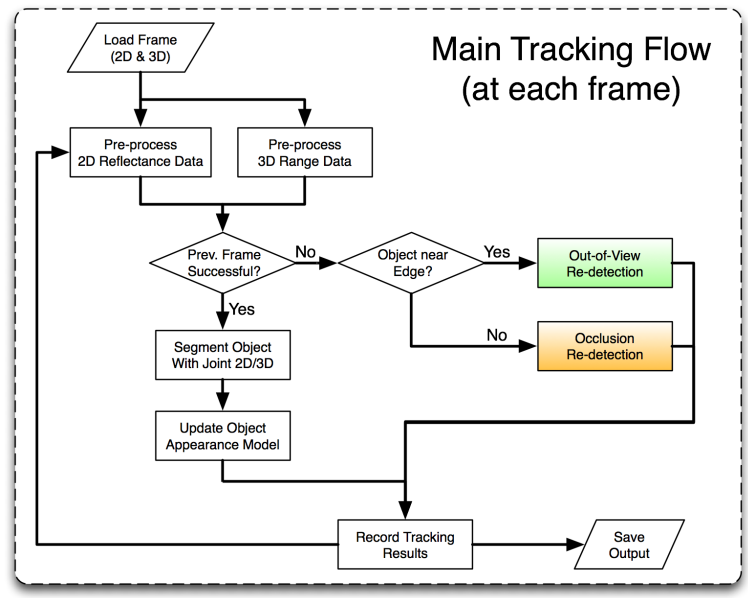


Figure 36: Flow chart describing overall operation of the tracker.

measures. Thus, μ_{in}, μ_{out}, Σ_{in}, and Σ_{out} are each vector valued. Consequently, G_{in} and G_{out} each map $\mathbb{R}^2 \to \mathbb{R}$. Unlike most segmentation techniques, this statistical information is not used directly, but rather is used indirectly to create a shape model

$$S(x) = H_{\epsilon_2}\left[\log\left(G_{\text{in}}(I(x))\right) - \log\left(G_{\text{out}}(I(x))\right)\right], \tag{54}$$

which serves as a labeling function with ϵ_2 being the threshold for our "estimated" shape. The image shape model S is the most likely shape of the object given current statistical estimates. Note that to mitigate the addition of another parameter, ϵ_2 is simply a scalar multiple of ϵ. From this, the segmenting curve is driven towards this shape by minimizing the following energy

$$E_{\text{image}}(\phi) = \|H\phi - S\|^2 = \frac{1}{2}\int_{\Omega}(H\phi(x) - S(x))^2 dx. \tag{55}$$

Specifically, this energy is the L_2 distance between the shape of the current segmentation and the current estimate of the "correct" shape. Using the calculus of variations,

we are then able to compute the gradient $\nabla_\phi \mathrm{E}_{\mathrm{image}}$ as follows:

$$\begin{aligned}\nabla_\phi \mathrm{E}_{\mathrm{image}} = \delta\phi.(H\phi(x) - S(x)) + \beta_\mu^{\mathrm{out}}.\nabla_\phi \mu_{\mathrm{out}} + \beta_\Sigma^{\mathrm{out}}.\nabla_\phi \Sigma_{out} \\ - \beta_\mu^{\mathrm{in}}.\nabla_\phi \mu_{\mathrm{in}} - \beta_\Sigma^{\mathrm{in}}.\nabla_\phi \Sigma_{\mathrm{in}}\end{aligned} \tag{56}$$

where the expressions of the coefficients β_μ^{in} and $\beta_\Sigma^{\mathrm{in}}$ are given by

$$\begin{aligned}\beta_\mu^{\mathrm{in}} &= \int_\Omega \gamma(u)\Sigma_{\mathrm{in}}^{-1}(I(u) - \mu_{\mathrm{in}})du \\ \beta_\Sigma^{\mathrm{in}} &= \frac{1}{2}\int_\Omega \gamma(u).(\Sigma_{in}^1(I(u) - \mu_{\mathrm{in}})(I(u) - \mu_{\mathrm{in}})^T\Sigma_{\mathrm{in}}^{-1} - \Sigma_{in}^{-1})du\end{aligned} \tag{57}$$

with $\gamma(x) = (H\phi(x) - S(x)).\delta_{\epsilon_2}\big(\log(\frac{G_{\mathrm{in}}(I(x))}{G_{\mathrm{out}}(I(x))}\big)$. In particular, because we deal with 3D range data as well as 2D reflectance data, $\beta_\mu^{\mathrm{in}} \in \mathbb{R}^2$ and $\beta_\mu^{\mathrm{in}} \in \mathbb{R}^4$. Likewise, both β_μ^{out} and $\beta_\Sigma^{\mathrm{out}}$ can be computed by replacing μ_{in} with μ_{out} and Σ_{in} with Σ_{out} in Equation (57). The expression of the gradients for each of the statistical moments are

$$\begin{aligned}\nabla_\phi \mu_{\mathrm{in}} &= \delta\phi.\left(\frac{I - \mu_{\mathrm{in}}}{A_{\mathrm{in}}}\right) \quad , \quad \nabla_\phi \mu_{\mathrm{out}} = \delta\phi.\left(\frac{I - \mu_{\mathrm{out}}}{A_{\mathrm{out}}}\right) \\ \nabla_\phi \Sigma_{\mathrm{in}} &= \delta\phi.\left(\frac{(I - \mu_{\mathrm{in}})(I - \mu_{\mathrm{in}})^T - \Sigma_{\mathrm{in}}}{A_{\mathrm{in}}}\right) \\ \nabla_\phi \Sigma_{\mathrm{out}} &= \delta\phi.\left(\frac{(I - \mu_{\mathrm{out}})(I - \mu_{\mathrm{out}})^T - \Sigma_{\mathrm{out}}}{A_{\mathrm{out}}}\right)\end{aligned}$$

With the above results, we can now place the image fidelity term in the overall GAC scheme. That is, to minimize this energy via gradient descent, ϕ is updated at each iteration according to

$$\frac{d\phi}{dt} = -\nabla_\phi E_{\mathrm{image}} + \lambda\delta\phi \cdot \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{58}$$

where the second term in the right-hand-side acts as a regularizing term that penalizes high curvatures. We note that while other active contour energy and segmentation methodologies can be employed, the above scheme leverages the image shape model created from statistical information. This allows for more robust segmentations for 3DLADAR imagery, where under-segmentation is preferred as opposed to

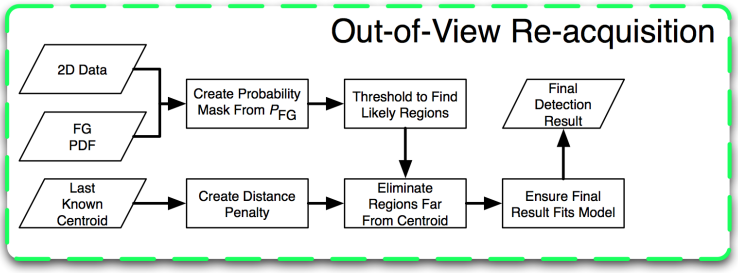


Figure 37: Flow chart describing the process for re-acquiring objects that have temporarily moved out of view.

over-segmentation. This preference is driven by the fact that each pixel identified as "on-target" in the final segmentation corresponds to a 3D point on the real target. Hence, an under-segmentation ensures that all "on-target" points can be used to understand the shape and pose of the target in world coordinates without being distracted by "off-target" background points.

5.1.2 Appearance Models

In a video sequence, we make the assumption of temporal coherency and assume that characteristics of the object will vary slowly over time. With this assumption, we employ appearance models to aid in pre-processing the data, detecting tracking failures, and re-acquiring lost objects.

Three primary aspects of the appearance models exist: Gaussian statistical models, probability density functions (PDFs), and shape information. Gaussian statistical models G_{in} and G_{out}, as described in Section 5.1.1, are stored at each iteration to aid in re-acquisition of the object when it is temporarily occluded. In addition, full intensity histograms of the reflectance data present in the object and background are scaled to produce PDFs P_{in} and P_{out} that are stored at each iteration. This allows the tracker to compare the PDFs of new segmentations with recent segmentations to detect tracking failures and subsequently re-acquire the object. PDFs are always

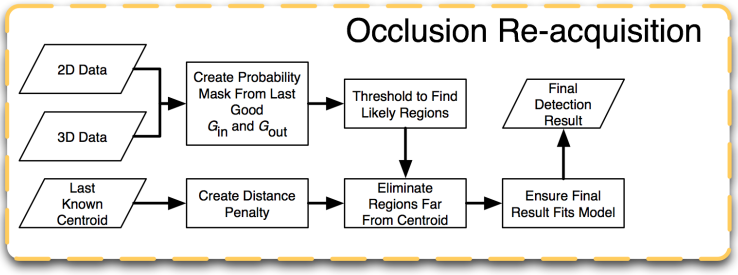


Figure 38: Flow chart describing the process for re-acquiring objects that have been temporarily occluded.

compared using the Bhattacharyya measure,

$$B\big(P_1(x), P_2(x)\big) = \sum_{x=0}^{k} \sqrt{P_1(x) \cdot P_2(x)}dx, \tag{59}$$

where P_1 and P_2 are any two PDFs [7]. The Bhattacharyya measure is a useful tool because it provides a scalar value in the range $[0, 1]$ corresponding to the similarity of the two PDFs. Finally, shape information such as the area (in pixels) of the object in recent frames are stored to help detect when the object is occluded or moving out of view.

5.2 Tracking Algorithm

In this section, we discuss the procedure that the tracker follows to continually estimate the 3D location of the object in the 3DLADAR sequence. As will be discussed, the tracker combines segmentation and heuristic detection in a robust tracking framework. In short, the 3DLADAR data is pre-processed and a joint 2D/3D segmentation is performed at each iteration to find all 3D data points on the target. If necessary, steps are taken to re-acquire lost objects.

5.2.1 Main Tracking Loop

This section introduces the various steps at a high level, and refers to the appropriate sections below where additional detail can be found. Figure 36 summarizes the algorithm, and provides a visual representation of the main tracking loop.

The first step is to load and pre-process the 3D range data as described in Section 5.2.2 and the 2D reflectance data as discussed in Section 5.2.3. These data make up the 3DLADAR imagery available at each frame, and both are required to robustly track the target. Pre-processing of the data prepares it for segmentation.

Next, a decision is made. If the tracker was successful in the previous frame, the tracker proceeds with segmentation as normal. If not, special measures are taken to re-acquire the target. Depending on the last known position of the object, one of two re-acquisition procedures are run.

If the object was last seen near the edge of the image, it is assumed that the object was lost because it moved out of the field of view (FOV). In that case, the procedure described in Section 5.2.4 is used to re-acquire the object based solely on reflectance data. Reflectance data is used alone in this case because the object's 3D range may change dramatically while missing from the FOV, but it's reflectance properties should remain constant.

If the object was last seen toward the center of the image, then the loss of track is attributed to occlusion by other objects in the scene. In this case a different procedure, described in Section 5.2.5, is used to reacquired the object based on both reflectance and range data.

In normal operation, when the previous frame was successfully tracked, the object is most likely still visible. In this case, a segmentation is performed to find the object's boundaries and the object's appearance model is updated. First, the segmentation algorithm described in Section 5.1.1 is initialized using the tracking result from the previous frame and allowed to run for several iterations. The segmentation process captures the shape of the object in the current frame and indicates which 3DLADAR data points fall within the object. Next, the object's appearance model is updated as described in Section 5.2.6. Finally, the procedure repeats with the loading of 3DLADAR data at time $t + 1$.

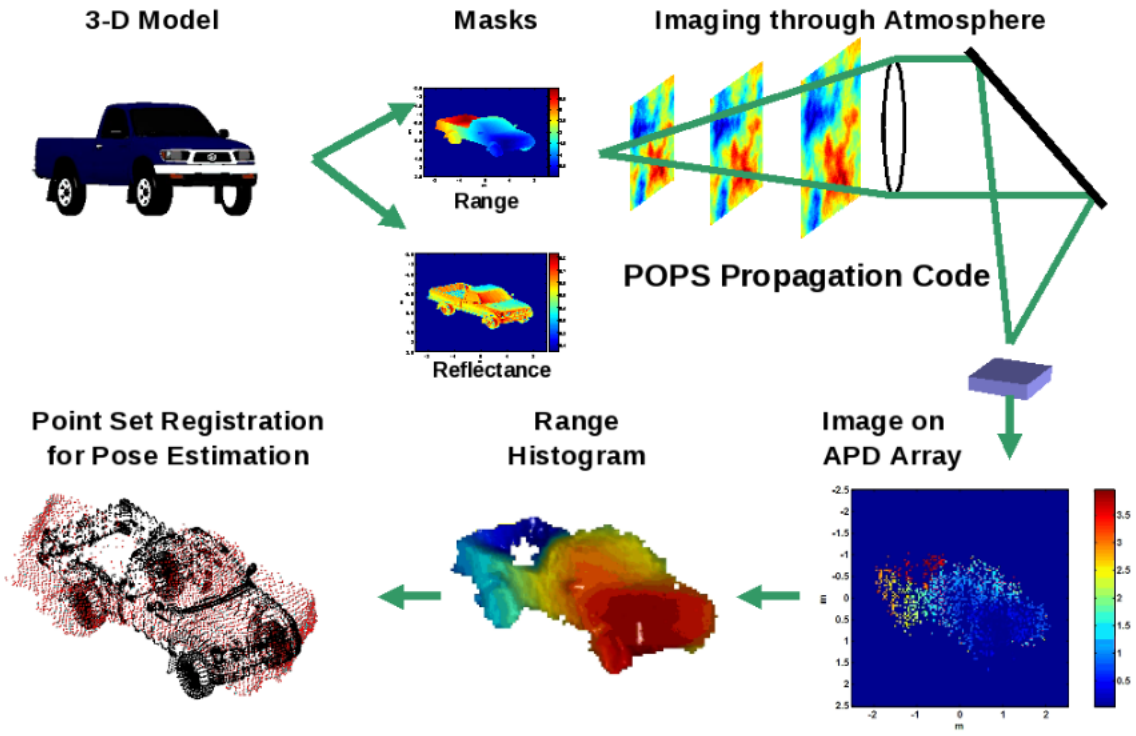


Figure 39: Simulated Environment for Producing 3DLADAR Imagery with Atmospheric Turbulence.

5.2.2 Pre-processing 3D Range Data

As mentioned in the introduction of this chapter, 3DLADAR offers the capability to resolve all spatial dimensions given the angle-angle and range information. However, unlike typical 2D systems, the notion of an "image" is not immediately available and an image must be formed based on specific assumptions about the acquisition of 3DLADAR data. Moreover, because the 3DLADAR imaging system is not fully developed, we demonstrate results on a simulated 3DLADAR environment, which is pictorially seen in Figure 39.

That is, the 3DLADAR platform delivers several short-pulsed lasers at specific instance of time in hopes of actively illuminating the target. For convenience, we label these series of pulses as a "frame." For each "frame," the corresponding photon arrival time is recorded by the focal plane of an avalanche photo-diode (APD) array.

In some cases, arrival times may never be recorded. For instance, if the laser pulse misses the target in clear sky and is never reflected back to the imaging platform. In any event, much of scene will not be recorded for a single frame due to poorly reflected photons. Luckily, the laser pulses are delivered at a pre-specified frequency that is generally much higher than real-time imaging systems.

From this, we are now able to define an "image" as a combination of frames that have been taken over a certain period. Each pixel value of the resulting image is formed by choosing the statistical mode of the values received throughout a series of frames at each pixel location. Additionally, because of the temporal coherency inherent to time-varying imagery, median filtering is performed temporally from image to image to mitigate artifacts caused from the imaging system.

5.2.3 Pre-processing 2D Reflectance Data

Processing of the 2D data is also important. Because the object and background may be multi-modal in the reflectance data, pre-processing ensures that the object is distinguishable from the background so that segmentation may proceed with ease. First, the PDF of the object (P_{in}) and background (P_{out}) are estimated using the reflectance data and the segmentation result from the previous frame. Next, a foreground histogram, P_{FG} is formed by removing intensity components of the background from intensity components of the foreground,

$$P_{\text{FG}} = \begin{cases} P_{\text{in}} - P_{\text{out}} & (P_{\text{in}} - P_{\text{out}}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{60}$$

and re-normalizing so that $\|P_{\text{FG}}\| = 1$. Finally, a new image is created corresponding to the likelihood that each pixel is a member of the foreground. Hence, the final result of 2D reflectance pre-processing is

$$\tilde{I}_{\text{ref}}(x) = P_{\text{FG}}(I_{\text{ref}}(x)). \tag{61}$$

5.2.4 Re-acquisition (Out-of-View)

If the 3DLADAR device can not accurately follow the movement of the object based on estimates from the tracker, the object may appear to move out of the image domain, and therefore out of the field of view. If this occurs, the procedure shown in Figure 37 is used to detect the object and re-acquire the track once it returns to the FOV.

When searching for the object after it has left view, only the reflectance data is utilized. This is because the object may change its 3D position dramatically during the time it is out of view. Conversely, the reflectance properties of the object should remain constant. Additionally, we assume that the object will reappear near the position (in image coordinates) that it was last seen. With these assumptions in mind, the procedure is as follows.

First, a probability map, $\mathbb{P}(x) = P_{\mathrm{FG}}(I_{\mathrm{ref}}(x))$ is created based on the foreground PDF from the last successfully tracked frame. This probability map will have a high value at pixels that have reflectance consistent with the object, but not the background. This probability map is then thresholded to produce a binary mask, which selects regions with an appearance similar to the object. Next, a distance penalty is employed to remove any candidate regions that are too far from the last known location of the object. If the resulting region has an appropriate size then it is accepted as the detection result and tracking will continue normally at the next iteration. Otherwise, this detection procedure is repeated at the next frame.

5.2.5 Re-acquisition (Occlusion)

Another failure mode of the tracker occurs when another object in the scene occludes the object of interest, blocking it from view. The procedure shown in Figure 38 is used to re-acquire after the occlusion.

Occlusions of this type are typically shorter and often portions of the object remain

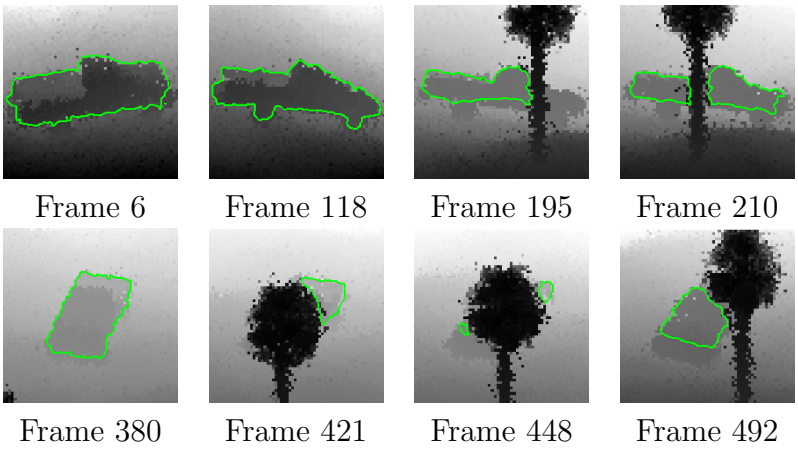


Figure 40: Visual tracking results of a truck passing through a noisy environment with several varying tree occlusions.

visible during most of the occlusion. Hence, we assume that the object will remain at a similar 3D depth and retain its 2D reflectance characteristics. For this case, both types of 3DLADAR data are used. Again, we assume that the object will reappear near the position (in image coordinates) that is was last seen.

To re-acquire the object, the shape model S described in Section 5.1.1 is constructed using the statistical models G_{in} and G_{out} from the last successful frame. This shape model is then thresholded to create a binary mask selecting candidate regions that may represent the object. Again, candidate regions that are far from the last known location of the object are excluded and the remaining region is assumed to be the object's current location. If the detected object has a size similar to the object when it was lost, a successful detection has occurred, and tracking will continue normally at the next iteration. Otherwise, this detection procedure will be repeated until the object is successfully re-acquired.

5.2.6 Target Appearance Model Update

After segmentation or re-acquisition has occurred successfully, the object's appearance model is updated. This process consists of adding current information such as P_{in}, size of the object (in pixels) and the (x, y) location in image coordinates of the object's

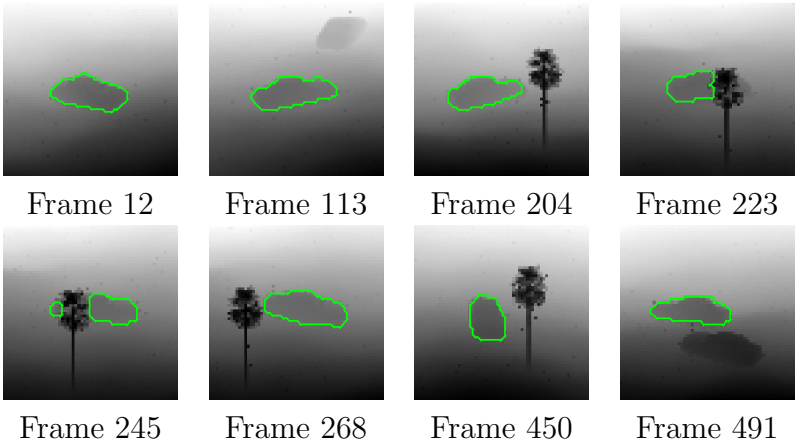


Figure 41: Visual tracking results of a car passing through a clutter environment that includes complete occlusions as well as non-cooperative targets.

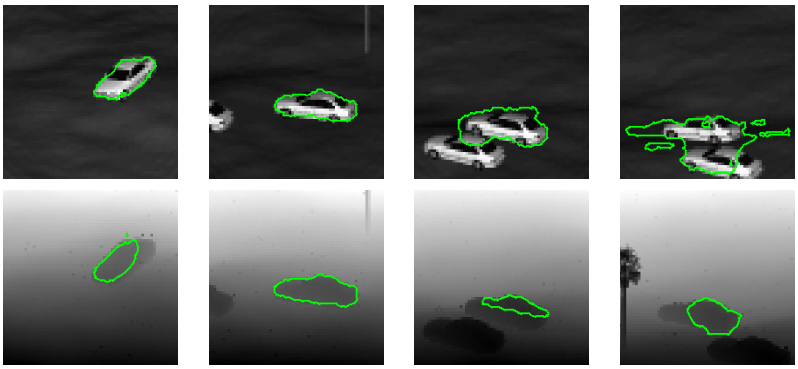
centroid to a history of recent values. These values are used to determine appropriate values and thresholds during pre-processing and detection for subsequent frames.

5.2.7 Registration and Pose Estimation

Now that segmentation has been performed, a point cloud estimate can be extracted using the specifications of the imaging 3DLADAR system (e.g., slant path, viewing angle). From this, the problem then becomes one of pose estimation or point set registration. In particular, we have experimented with both the ICP methodology as well a variant of ICP using particle filtering proposed in Chapter 3. Although the corresponding pose results were obtained at MIT-Lincoln Labs and are not present in this chapter, we should note the pose of the target remained sufficiently accurate throughout all tracking scenarios.

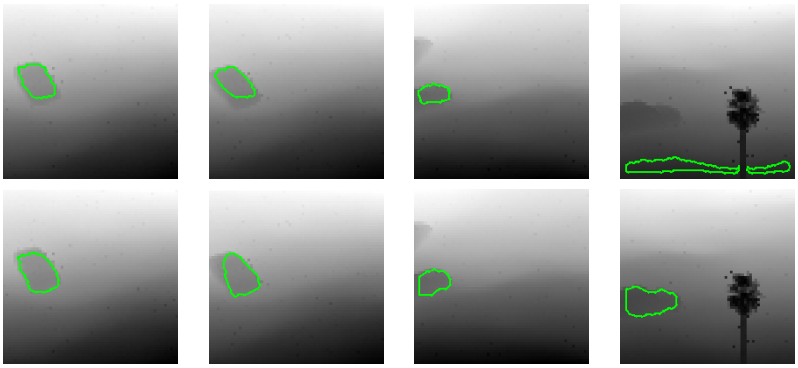
5.3 Experiments

In this section, we demonstrate the algorithm's robustness to turbulence, occlusions, clutter, and erratic behavior on several challenging video sequences. We also motivate the need to fuse both 2D reflectance and 3D range data. In particular, the sequences (both 2D and 3D) were simulated by MIT-Lincoln Labs for several degrees



Frame 700 Frame 783 Frame 808 Frame 822

Figure 42: Visual tracking results of a target car near a similar car. *Top Row*: Using only 2D reflectance data, the segmentation fails and leaks onto the nearby car. *Bottom Row*: Using combined reflectance and range data, the target car is tracked successfully.



Frame 113 Frame 156 Frame 209 Frame 342

Figure 43: Visual tracking results of a target car as it moves out of the field of view and re-enters. *Top Row*: Using only 3D range data, the detection fails and the system begins to track the background. *Bottom Row*: Using combined reflectance and range data, the target car is tracked successfully.

of turbulence and image sizes. However, in this chapter, we only present results for two levels of turbulence and image size of 64X64 pixels. We note that MIT-Lincoln Labs currently has an active 32X32 APD array in the field, but are able to simulate 32X32, 64X64, and 128X128 image sizes.

5.3.1 Robustness to Varying Views, Occlusions, and Turbulence

Let us begin with the first sequence of a truck that moves in open terrain. This is shown in Figure 40, where one can see that target occupies much of the field of view

(FOV). Additionally, significant turbulence and several occlusions by trees obfuscate the object's visibility with respect to the imaging camera. Nevertheless, the algorithm successfully tracks the truck throughout the entire sequence.

The next sequence of which several frames are shown in Figure 41, demonstrates the algorithm's robustness to complete occlusions as well as a clutter environment that arises from non-cooperative objects moving around within the scene. In particular, another car that is identical in appearance (but not in depth), passes close to the target of interest. However, the algorithm successfully tracks the car throughout the sequence using the proposed tracking technique.

5.3.2 Benefits of Coupling 2D and 3D Information

In Figure 42 and Figure 43, we demonstrate inherent problems that might occur when tracking with purely 2D reflectance imagery or 3D range imagery on a challenging scenario. While similar to the sequence presented in Figure 41, here the imaging camera in this sequence loses track of the vehicle as it begins linger near the edge of the image before visibility is completely lost. These experiments show that tracking and detection can be performed by leveraging both 2D and 3D information (as opposed to tracking with reflectance or depth alone).

For example, if non-cooperative targets such as two cars are present in a particular scene, it becomes increasingly difficult to distinguish the target of interest from a statistical point of view when using purely 2D reflectance. In the top row of Figure 42, we see that the active contour leaks onto the second car when the two approach each other. However, when we include 3D depth information, the target can be successfully distinguished. This is shown on the bottom row of the same figure.

Moreover, Figure 43 presents the major drawback associated with using range information alone. In the case of erratic behavior, whereby the object leaves the image view, detection becomes unreliable. That is, when the truck leaves the FOV,

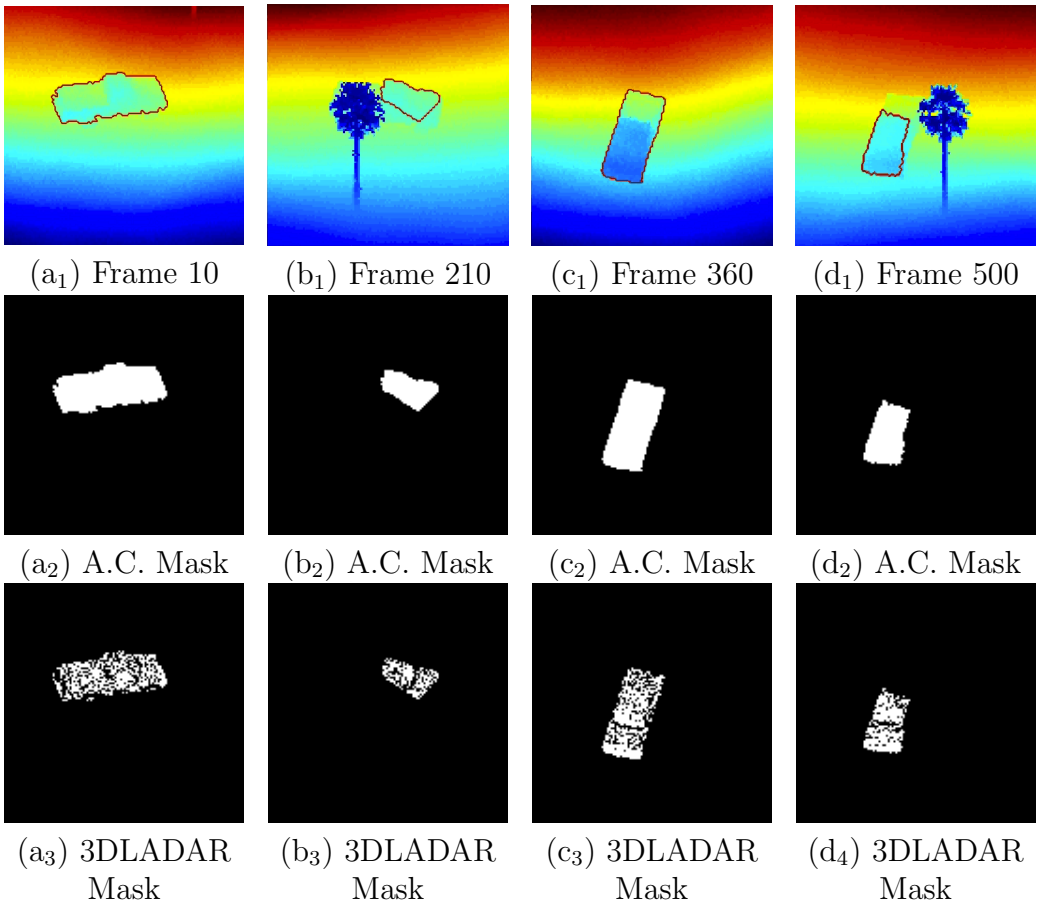
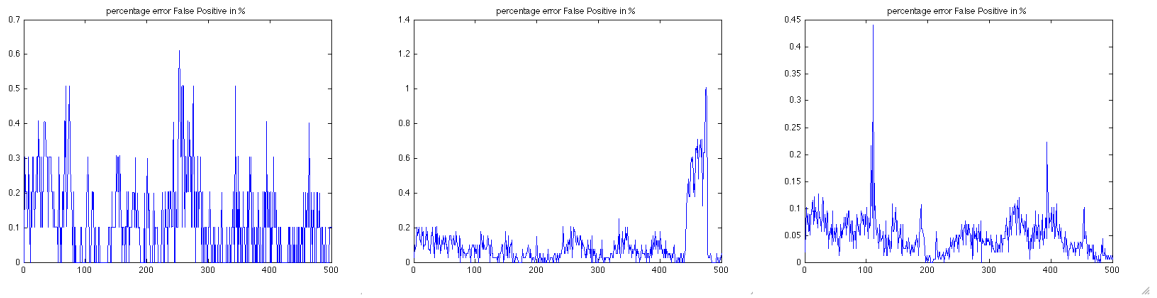


Figure 44: Tracking Sequence of a Moving Target with Corresponding Binary Masks Provided by Active Contour and 3DLADAR Masks. Top Row: Tracked 3DLADAR Images. Middle Row: Binary Masks Associated with Active Contour (A.C.). Bottom Row: 3DLADAR Filtered Mask for Points Only Registered By System

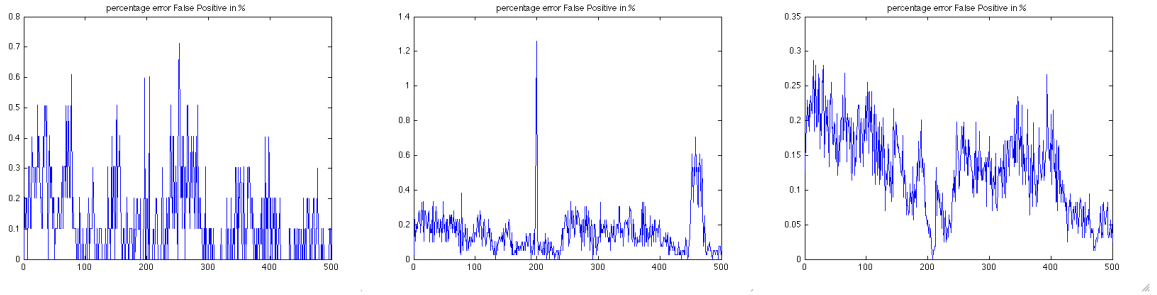
it will have a specific depth value, but when it re-enters its depth value may be completely different. Unfortunately, in such a case, the model for reacquisition is no longer valid and detection will fail. By leveraging on 2D information, we are able to detect and re-acquire in a more reliable fashion.

5.3.3 Quantifying Tracking Results with Ground Truth

Until now, we have provided qualitative tracking results for several difficult sequences, but have yet compared this to ground truth available by the simulator. In this section, we revisit the important notion of being able to under-segment the "target" such that pixels captured by the active contour contain relatively few or no background



(a_1) False Positive: 32X32 No Turbulence (b_1) False Positive: 64X64 No Turbulence (c_1)False Positive: 128X128 No Turbulence

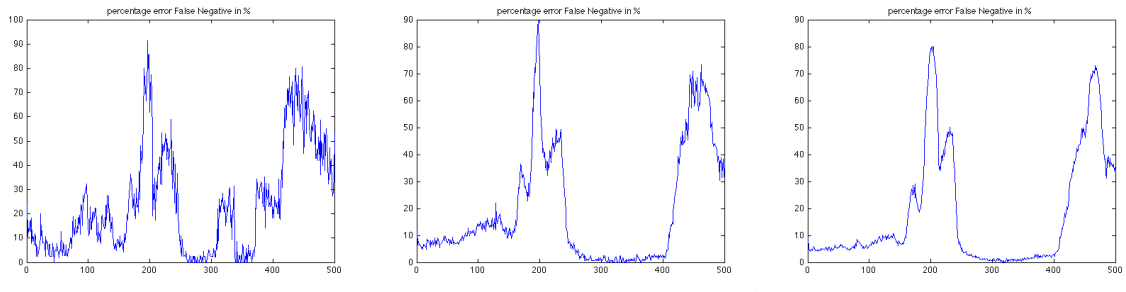


(a_2) False Positive: 32X32 High Turbulence (b_2) False Positive: 64X64 High Turbulence (c_2)False Positive: 128X128 High Turbulence

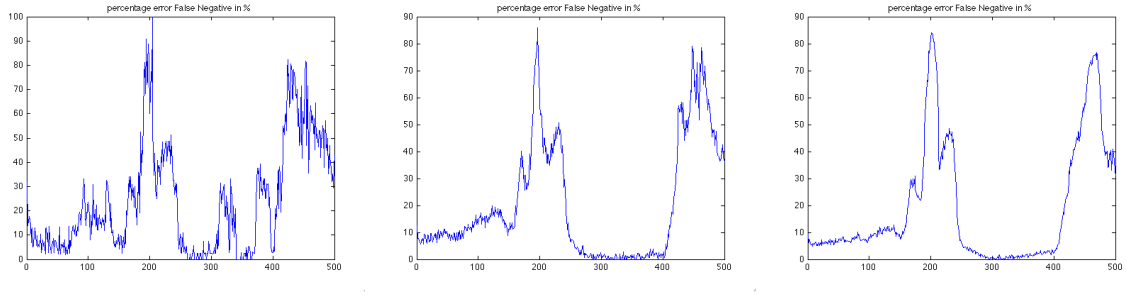
Figure 45: Quantifying Segmentation Results with Ground Truth via False Positives. Top Row: False Positives for Image Sizes of 32X32, 64X64 and 128X128 with No Turbulence. Bottom Row: False Positives for Image Sizes of 32X32, 64X64 and 128X128 with High Turbulence. **Note: Scales are different for each image so that one can see small deviations in tracking.**

features at the cost of fewer target pixels. Thus, we consider the ability to capture background pixels as a "False Positive" while missing target pixels will be denoted as "False Negatives."

Several images of the sequence in which we quantify our tracking algorithm is shown in Figure 44. In particular, we would like to point out that in order for one to compare the ground truth with the tracked result as well as being able to properly register the extracted 3D point set, one must first only examine those points registered by the 3DLADAR imaging system. That is, during the pre-processing step one must artificially fill in certain regions of the image where the avalanche photo-diode array did not receive a photon count or image value. This again could be due to perhaps imaging the sky where the arrival time is infinite (not reflected). In turn, the visual



(a_1) False Negative: 32X32 No Turbulence (b_1) False Negative: 64X64 No Turbulence (c_1)False Negative: 128X128 No Turbulence



(a_2) False Negative: 32X32 High Turbulence (b_2) False Negative: 64X64 High Turbulence (c_2)False Negative: 128X128 High Turbulence

Figure 46: Quantifying Segmentation Results with Ground Truth via False Negatives. Top Row: False Negatives for Image Sizes of 32X32, 64X64 and 128X128 with No Turbulence. Bottom Row: False Negatives for Image Sizes of 32X32, 64X64 and 128X128 with High Turbulence. **Note: Scales are different for each image so that one can see small deviations in tracking.**

images seen in this chapter have been pre-processed so that they are "smooth" and continuous throughout each location in the x-y image plane. Thus, when extracting "target" features, we only extract those points that were initially registered by the imaging system. These filtered masks and points are shown in the bottom row of Figure 44.

Consequently, we are now able to compare the filtered binary masks with that of the ground truth. Interestingly, we see in Figure 45 that our False Positives are very low (below 1%) for each frame in each sequence of differing image sizes and turbulence. Again, this should be particularly important since we will be returning only "target" pixels and should then ease the applicability of a point set registration algorithm to estimate the target's pose.

In regards to the False Negative's, we tend to have a higher percent error when facing occlusion as shown in Figure 46. That is, we are not able to capture as much of the target as we would like to. However, we still maintain track throughout the sequence and do indeed recover from the tree occlusion as seen in Figure 44. More importantly, the target pixels that we do retain, when facing occlusions, are important features for the ICP-like algorithms. Unfortunately, this detail is not visible in the results shown. Ideally, point set registration can be performed if one is given (few) quality features of the target.

5.4 Chapter Conclusion

In this chapter, we have described a robust tracking algorithm capable of continuously tracking a target in 3DLADAR imagery despite the presence of noise, turbulence, occlusions, and temporary motion of the target out of the field of view. This is accomplished by combining geometric active contours and reasonable heuristics, which are used to re-acquire the target if tracking is disrupted. We also presented experiments to demonstrate the algorithm's robustness to turbulence, occlusions, clutter, and erratic behavior on several challenging video sequences.

CHAPTER VI

CONCLUDING REMARKS AND FUTURE RESEARCH

In this thesis, we proposed an algorithm that jointly performs 2D image segmentation and 3D pose estimation for a general class of 3D rigid objects. This was done by first introducing a novel distribution metric for image segmentation along with a point set registration that employs a particle filter to do pose estimation. Lastly, to demonstrate the viability of the algorithms developed in this thesis, we presented their application to the real world setting in the form of visual tracking of 3DLADAR imagery.

Specifically, in Chapter 2, we revisited a distribution metric that arises as a result from prediction theory where one can quantify the "distance" between two probability distributions as the standard deviation of the difference between logarithms of those distributions. From this, we then proposed an energy of similar form using level sets to perform image segmentation. That is, by making the assumption that the object of interest is statistically different from the background, we opted to maximize the distance, defined by the distribution metric, between their corresponding distributions. Experimental results were demonstrated on several types of challenging low contrast images.

Shifting our attention to an opposite, but related field of pose estimation, Chapter 3 developed a particle filtering point set registration algorithm. In particular, we provided a solution to the problem of registering two point sets that differ by a rigid body transformation. Treating the pose parameters as hidden Markov random variables, we naturally incorporated a particle filter whereby the correspondences established at an artificial time t was viewed as "information" received online. More

importantly, this allowed us to characterize the variability in the overall registration process, which in turn was exploited within the algorithm.

After laying the fundamentals of both image segmentation and pose estimation, Chapter 4 attempted to bridge the gap between these two areas for the specific, but general class of 3D rigid objects. This was done by relying on surface differential geometry and projective geometry to link certain intrinsic details of both the 2D and 3D world. As a result, we proposed a unique energy functional that accomplishes both tasks of segmentation and pose estimation without the need for point correspondences or specific constraints on the type of 3D shape. More importantly, we utilized nonlinear manifold learning techniques for a general class of objects or deformations for which one was not able to associate a skeleton model.

Lastly, in Chapter 5 we demonstrated the importance of the proposed algorithms by presenting their applicability in the context of tactical tracking for 3DLADAR imagery. Using a two pronged approach of employing both segmentation and pose estimation, the resulting tracking framework was able to handle challenging situations where the "target" was obscured from view due to erratic motion, occlusions, and turbulence.

We should note that while strides have been made in both image segmentation and pose estimation, the direction of future research is numerous. In particular, the algorithm developed in Chapter 3 only attempts to solve pose estimation for a rigid body transformation. However, the relationship or transformation between two objects is usually more complex. For example, the neural fiber bundles or tracts that can be found in diffusion weighted magnetic resonance imagery (DW-MRI) can not be explained by simply a rigid or affine transformation. Thus, one future direction for research is to explore a poly-affine transformation in which the overall transformation can be seen as multiple affine transformations operating simultaneously in conjunction. Another avenue for a future research direction would be to extend the algorithm

in Chapter 4 so that it accounts for dynamics that are found in video sequences. That is, one could employ a particle filter like of that Chapter 3 or a Kalman-based filter to jointly perform 2D image segmentation and 3D pose estimation. Finally, we should note that the tracking framework presented in Chapter 5 is for the most part considered to be "open-loop." In order to achieve a more robust tracking framework, one should try to "close the loop."

APPENDIX A

MINIMIZATION OF THE PROPOSED "LOCAL" OPTIMIZER FOR REGISTRATION

We first linearize both translation and rotation:

$$\mathbf{t} = c_1\mathbf{t_x} + c_2\mathbf{t_y} + c_3\mathbf{t_z}$$

$$\mathbf{R} \approx I + c_1\mathbf{R_x} + c_2\mathbf{R_y} + c_3\mathbf{R_z} \qquad \text{where}$$

$$\mathbf{R_x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \mathbf{R_y} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{R_z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then reorganizing terms gives us:

$$\begin{aligned} & n_i^T[d_i - \mathbf{R}m_i - \mathbf{t}] \\ \approx & n_i^T\left(d_i - m_i - c_1\mathbf{R_x}m_i - c_2\mathbf{R_y}m_i - c_3\mathbf{R_z}m_i - c_4\mathbf{t_x} - c_5\mathbf{t_x} - c_6\mathbf{t_x}\right) \\ \approx & \Delta_i - \tilde{m}_i^T C \end{aligned}$$

where

$$\Delta_i = n_i^T(d_i - m_i)$$

$$\tilde{m}_i = \left[n_i^T\mathbf{R_x}m_i \;\; n_i^T\mathbf{R_y}m_i \;\; n_i^T\mathbf{R_z}m_i \;\; n_i^T\mathbf{t_x} \;\; n_i^T\mathbf{t_y} \;\; n_i^T\mathbf{t_z}\right]^T$$

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{bmatrix}$$

Now we can re-write the above energy functional as

$$E = \lambda \sum_{i=1}^{N_d} \omega_i \cdot \exp\left(-\frac{1}{4}(\Delta_i - \tilde{m}_i^T C)^T (\Delta_i - \tilde{m}_i^T C)\right)$$
$$= \lambda \sum_{i=1}^{N_d} \omega_i \cdot \exp\left(-\frac{1}{4}(\Delta_i^T \Delta_i - C^T \tilde{m}_i \Delta_i - ... \right.$$
$$\left. \Delta_i^T \tilde{m}_i^T C + C^T \tilde{m}_i \tilde{m}_i^T C)\right)$$

Taking the variation w.r.t. to C yields

$$\frac{\partial E}{\partial C} = \lambda \sum_{i=1}^{N_d} \omega_i \cdot \left(-\frac{1}{4}(-2\tilde{m}_i \Delta_i + 2\tilde{m}_i \tilde{m}_i^T C)\right) \cdot ...$$
$$\exp\left(-\frac{1}{4}(\Delta_i - \tilde{m}_i^T C)^T (\Delta_i - \tilde{m}_i^T C)\right)$$

$$\frac{\partial E}{\partial C} = \sum_{i=1}^{N_d} \underbrace{\omega_i}_{weight} \cdot \overbrace{(\tilde{m}_i \Delta_i - \tilde{m}_i \tilde{m}_i^T C)}^{\text{ICP}} \cdot ...$$
$$\underbrace{\exp\left(-\frac{1}{4}(\Delta_i - \tilde{m}_i^T C)^T (\Delta_i - \tilde{m}_i^T C)\right)}_{\text{Penalizes Outliers}}$$

APPENDIX B

GRADIENT DERIVATION OF EXPONENTIAL KERNEL PRE-IMAGE

If we let

$$\mathcal{J}_i = \tilde{\gamma}_i\big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\big) \quad \text{and} \quad \eta_i = \frac{\partial \mathcal{J}_i}{\partial \omega_n}$$

then the general gradient form is

$$\frac{\partial \hat{\psi}}{\partial \omega_n} = \frac{\sum_i^N \eta_i \cdot \psi_i}{\sum_i^N \mathcal{J}_i} - \frac{\big(\sum_i^N \eta_i\big)\big(\sum_i^N \mathcal{J}_i \cdot \psi_i\big)}{(\sum_i^N \mathcal{J}_i)^2}$$

Thus, taking the derivative of $\mathcal{J}_i$ w.r.t to ω_i yields

$$\eta_i = \underbrace{\nabla_{\omega_n}\tilde{\gamma}_i}_{(a)} \cdot \big(1 - \frac{1}{2}d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\big) - \frac{1}{2}\tilde{\gamma}_i \cdot \underbrace{\nabla_{\omega_n} d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))}_{(b)}$$

Now taking the derivative of (a) above by using Equation (36), we get

$$\tilde{\gamma}_i = \sum_n^l \frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} + \frac{1}{N}\left(1 - \sum_j^N \sum_n^l \frac{\omega_n u_{nj}}{\sqrt{\lambda_n}}\right)$$

$$\nabla_{\omega_n}\tilde{\gamma}_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}}$$

Next, we compute the gradient of part (b). Recall that

$$d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i)) = \|P^l\varphi(\psi)\|^2 + \|\varphi(\psi_i)\|^2 - 2\langle P^l\varphi(\psi), \varphi(\psi_i)\rangle$$

From this, we can begin to express each term's dependency on the shape weight ω_n. The first term can be expressed as

$$\begin{aligned}
\|P^l\varphi(\psi)\|^2 &= \big(\sum_n^l \omega_n V_n + \bar{\varphi}\big)^T \big(\sum_n^l \omega_n V_n + \bar{\varphi}\big) \\
&= \big(\sum_n^l \sum_n^l \omega_n V_n^T V_n \omega_n\big) + \bar{\varphi}^T\bar{\varphi} + \big(2\bar{\varphi}^T \sum_n^l \omega_n V_n\big) \\
&= \big(\sum_n^l \omega_n^2\big) + \bar{\varphi}^T\bar{\varphi} + \big(2\bar{\varphi}^T \sum_n^l \omega_n V_n\big) \\
&= \big(\sum_n^l \omega_n^2 + 2\frac{\omega_n}{\sqrt{\lambda_n}}\bar{\varphi}^T\tilde{\phi}u_n\big) + \bar{\varphi}^T\bar{\varphi}
\end{aligned}$$

Similarly, the third term can be expressed as

$$\begin{aligned}
\langle P^l\varphi(\psi), \varphi(\psi_i)\rangle &= \Big(\sum_n^l \omega_n V_n + \bar{\varphi}\Big)^T \varphi(\psi_i) \\
&= \Big(\sum_n^l \big(\omega_n V_n\big)^T \varphi(\psi_i)\Big) + \bar{\varphi}^T\varphi(\psi_i) \\
&= \Big(\sum_n^l \omega_n (\frac{1}{\sqrt{\lambda_n}}\tilde{\phi}u_n)^T \varphi(\psi_i)\Big) + \bar{\varphi}^T\varphi(\psi_i) \\
&= \Big(\sum_n^l \frac{\omega_n}{\sqrt{\lambda_n}} (u_n^T\tilde{\phi}^T\varphi(\psi_i))\Big) + \bar{\varphi}^T\varphi(\psi_i)
\end{aligned}$$

Combining these terms then gives us

$$\begin{aligned}
d_{\mathcal{H}}^2 = &\Big(\sum_n^l \omega_n^2 + 2\frac{\omega_n}{\sqrt{\lambda_n}}(\tilde{\phi}^T\bar{\varphi})^T u_n - 2\frac{\omega_n}{\sqrt{\lambda_n}}(\tilde{\phi}^T\varphi(\psi_i))^T u_n\Big) \\
&-2\bar{\varphi}^T\varphi(\psi_i) + \bar{\varphi}^T\bar{\varphi} + \|\varphi(\psi_i)\|^2
\end{aligned}$$

Now, taking the derivative w.r.t ω_n, we arrive at the following:

$$\begin{aligned}
&\frac{d(d_{\mathcal{H}}^2)}{d\omega_n} = 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}} \cdot \big(\tilde{\phi}^T\bar{\varphi} - \tilde{\phi}^T\varphi(\psi_i)\big)^T u_n\Big) \\
&= 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}} \cdot \big((\phi - \bar{\phi})^T\bar{\varphi} - (\phi - \bar{\phi})^T\varphi(\psi_i)\big)^T u_n\Big) \\
&= 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}} \cdot \big(\phi^T\bar{\varphi} - \bar{\phi}^T\bar{\varphi} - \phi^T\varphi(\psi_i) + \bar{\phi}^T\varphi(\psi_i)\big)^T u_n\Big) \\
&= 2\Big(\frac{1}{\sqrt{\lambda_n}}\big(\frac{1}{N}\mathbf{Kl} - \frac{1}{N^2}(\mathbf{l}^T\mathbf{Kl})\mathbf{l} - \mathbf{k}_{\varphi_\mathbf{i}} + \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\big)^T u_n + \omega_n\Big)
\end{aligned}$$

where

$$\mathbf{k}_{\varphi_\mathbf{i}} = [k_{\varphi_\sigma}(\psi_i, \psi_1), \ldots, k_{\varphi_\sigma}(\psi_i, \psi_N)]^T$$

$$\phi = [\varphi(\psi_1), \varphi(\psi_2), ..., \varphi(\psi_N)] \qquad \text{and} \qquad \bar{\phi} = \phi - \tilde{\phi}.$$

Putting all of this together, we arrive at the desired gradient.

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(\psi), \varphi(\psi_i))\big) + ...$$
$$\frac{\tilde{\gamma}_i}{\sqrt{\lambda_n}}\big(\frac{1}{N^2}(\mathbf{l}^T\mathbf{K}\mathbf{l})\mathbf{l} - \frac{1}{N}\mathbf{K}\mathbf{l} + \mathbf{k}_{\varphi_\mathbf{i}} - \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\big)^T u_n - \omega_n.$$

We now have all of the components necessary to compute the overall shape gradient used in Chapter 4.

REFERENCES

[1] "Stanford University Computer Graphics Laboratory: The Stanford 3D Scanning Repository.,"

[2] Albota, M., Aul, B., Fouche, D., Heinrichs, R., Kocher, D., Marino, R., Mooney, J., Newbury, N., O'Brien, M., Player, B., Willard, B., and Zayhowski, J., "Three-Dimensional Imaging Laser Radars with Geiger-Mode Avalance Photodiode Arrays," *Lincoln Laboratory Journal*, no. 2, pp. 351–370, 2002.

[3] Armari, S., "Differential-geometrical methods in statistics, lecture notes in statistics," 1985.

[4] Balan, A., L.Sigal, Black, M., Davis, J., and Haussecker., H., "Detailed human shape and pose from images," in *CVPR*, 2007.

[5] Besl, P. J. and McKay, N. D., "A Method for Registration of 3-D Shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[6] Betser, A., Vela, P., and Tannenbaum, A., "Automatic tracking of flying vehicles using geodesic snakes and kalman filtering," in *Conference on Decision and Control*, pp. 1649–1654, December 2004.

[7] Bhattacharyya, A., "On a measure of divergence between two statistical populations defined by their probability distributions," *Calcutta Math. Soc.*, vol. 35, pp. 99–110, 1943.

[8] Blake, A. and Isard, M., eds., *Active Contours*. Springer, 1998.

[9] Blanz, V. and Vetter, T., "A Morphable Model for the Synthesis of 3d Faces," in *SIGGRAPH*, pp. 187–194, 1999.

[10] Bowden, R., Mitchell, T., and Sarhadi, M., "Reconstructing 3d Pose and Motion from a Single Camera View," in *Proceedings of the British Machine Vision Conference*, vol. 2, pp. 904–913, 1998.

[11] Bray, M., Kohli, P., and Torr, P., "Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts," in *ECCV*, pp. 642–655, 2006.

[12] Campbell, L., "The relation between information theory and differential geometry approach to statistics," vol. 35, pp. 199–210, 1985.

[13] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," in *IJCV*, vol. 22, pp. 61–79, 1997.

[14] Chan, T. and Vese, L., "Active contours without edges," *IEEE TIP*, vol. 10, no. 2, pp. 266–277, 2001.

[15] Charpiat, G., Faugeras, O., and Keriven, R., "Shape Statistics for Image Segmentation with Prior," in *CVPR*, pp. 1–6, 2007.

[16] Chen, Y., Huang, T., and Rui, Y., "Parametric contour tracking using unscented kalman filter," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 613–616., 2002.

[17] Chen, Y. and Medioni, G., "Object Modeling by Registration of Multiple Range Images," *Image Vision and Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[18] Chui, H. and Rangarajan, A., "A New Algorithm for Non-Rigid Point Matching," in *CVPR*, vol. 2, pp. 44–51, 2000.

[19] Chui, H., Rangarajan, A., Zhang, J., and Leonard, C. M., "Unsupervised Learning of An Atlas from Unlabeled Point-sets," *PAMI*, vol. 26, no. 2, pp. 160–172, 2004.

[20] Cover, T. and Thomas, J., *Elements of Information Theory*. New York: Wiley, 1991.

[21] Cremers, D., Kohlberger, T., and Schnoerr, C., "Shape statistics in kernel space for variational image segmentation," in *Pattern Recognition*, pp. 1292–1943, 2003.

[22] Dambreville, S., "Statistical and Geometric Methods for Shape-Driven Segmentation and Tracking," 2008.

[23] Dambreville, S., Rathi, Y., and Tannenbaum, A., "A framework for image segmentation using shape models and kernel space shape priors," *TPAMI*, vol. 30, no. 8, pp. 1385–1399, 2008.

[24] Dambreville, S., Sandhu, R., Yezzi, A., and Tannenbaum, A., "Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior," in *ECCV*, pp. 169–182, 2008.

[25] Debreuve, E., Gastaud, M., Barlaud, M., and Aubert, G., "Using the Shape Gradient for Active Contour Segmentation: From the Continuous to the Discrete Formulation," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 47–66, 2007.

[26] Deutscher, J. and Reid, I., "Articulated body motion capture by stochastic search," in *IJCV*, 2004.

[27] Dey, T. and Sun, J., "Normal and Feature Estimations from Noisy Point Clouds," in *Foundations of Software Technology and Theoretical Computer Science*, vol. 7, pp. 21–32, 2006.

[28] Dhome, M., Richetin, M., and Lapreste, J.-T., "Determination of the attitude of 3d objects from a single perspective view," *TPAMI*, vol. 11, no. 12, pp. 1265–1278, 1989.

[29] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*. New York: Springer Verlag, 2001.

[30] Drummond, T. and Cipolla, R., "Real-time tracking of multiple articulated structures in multiple views," in *ECCV*, 2000.

[31] Fitzgibbon, A. W., "Robust registration of 2D and 3D point sets," *Image Vision and Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[32] Freedman, D. and Zhang, T., "Active contours for tracking distributions," *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004.

[33] Gao, Y., Sandhu, R., Fichtinger, G., and A.Tannenbaum, "A coupled global registration and segmentation framework with application to magnetic resonance prostate imagery," *IEEE Transactions on Medical Imaging*, 2010.

[34] Gelfand, N., Mitra, N. J., Guibas, L. J., and Pottmann, H., "Robust Global Registration," in *Proc. Symp. Geom. Processing*, vol. 255, pp. 197–207, 2005.

[35] Georgiou, T., "Distances and riemannian metrics for spectral density functions," *IEEE Trans. on Signal Processing*, no. 8, pp. 3395–4003, 2007.

[36] Georgiou, T., "An intrinsic metric for power spectral density functions," *IEEE Trans. on Signal Processing Letters*, no. 8, pp. 561–563, 2007.

[37] Goldberger, J., Gordon, S., and .Greenspan, H., "An Efficient Image Similarity Measure Based on Approximations of KL-Divergence between Two Gaussian Mixtures," in *ICCV*, pp. 487–493, 2003.

[38] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing.* Addison-Wesley Longman., 2001.

[39] Gordon, N., Salmond, D., and Smith, A., "Novel Approach to Nonlinear/Nongaussian Bayesian State Estimation," *IEEE Proceedings on Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[40] Granger, S. and Pennec, X., "Multi-Scale EM-ICP: A Fast and Robust Approach for Surface Registration," in *ECCV*, vol. 2353, pp. 418–432, 2002.

[41] Han, F. and Zhu, S., "Bayesian Reconstruction of 3d Shapes and Scenes from a Single Image," in *Proc. Intl Workshop on High Level Knowledge in 3D Modeling and Motion*, vol. 2, 2003.

[42] Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[43] Hellinger, E., "Neaue begr undung der theorie der quadratischen formen von unendlichen vielen ver anderlichen," vol. 136, pp. 210–271, 1909.

[44] Horn, B., "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America Association*, vol. 4, no. 1, pp. 629–634, 1987.

[45] Hu, W., Tan, T., Wang, L., and Maybank, S., "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[46] Huber, D. F. and Hebert, M., "Fully Automatic Registration of Multiple 3D Data Sets," *Image Vision and Computing*, vol. 21, no. 7, pp. 637–650, 2003.

[47] Huber, P. J., *Robust Statistics*. New York: John Wiley & Sons, 1981.

[48] Jian, B. and Vemuri, B. C., "A Robust Algorithm for Point Set Registration Using Mixture of Gaussians," in *ICCV*, vol. 2, pp. 1246–1251, 2005.

[49] Johnson, A. and Herbert, M., "Using Spin-Images for Efficient Object Recogntion in Cluttered 3-D Scenes," *PAMI*, vol. 21, no. 5, pp. 433–449, 1999.

[50] Julier, S. and Uhlmann, J., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–420., 2004.

[51] Kass, A. and Terzopoulos, D., "Snakes: active contour models," *IJCV*, pp. 321–331, 1987.

[52] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., "Conformal curvature flows: From phase transitions to active vision," *Arch. Ration. Mech. Anal.*, vol. 134, pp. 275–301, Sept. 1996.

[53] Kohli, P., Rihan, J., Bray, M., and Torr, P., "Simultaneous segmentation and 3d pose estimation of humans using dynamic graph cuts," *IJCV*, vol. 79, no. 3, pp. 285,298, 2008.

[54] Kwok, J. and Tsang, I., "The pre-image problem in kernel methods," in *IEEE transactions on neural networks*, vol. 15, pp. 1517–1525, 2004.

[55] Leventon, M., Grimson, E., and Faugeras, O., "Statistical shape influence in geodesic active contours," in *Proc. IEEE CVPR*, pp. 1316–1324, 2000.

[56] Ma, B. and Ellis, R. E., "Surface-Based Registration with a Particle Filter," in *MICCAI*, vol. 3216, pp. 566–573, 2004.

[57] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S., *An invitation to 3D vision.* Springer.

[58] Makadia, A., IV, A. P., and Daniilidis, K., "Fully Automatic Registration of 3D Point Clouds," in *CVPR*, vol. 1, pp. 1297–1304, 2006.

[59] Marchand, E., Bouthemy, P., and Chaumette, F., "A 2d-3d model-based approach to real-time visual tracking," *IVC*, vol. 19, no. 13, pp. 941–955, 2001.

[60] Mercer, J., "Functions of positive and negative type and their connection with the theory of integral equations," in *Philos. Trans. Roy. Soc. London*, vol. 209, pp. 415–446, 1909.

[61] Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., and Rätsch, G., "Kernel PCA and de–noising in feature spaces," in *Advances in Neural Information Processing Systems*, vol. 11, MIT Press, 1999.

[62] Moghari, M. and Abolmaesumi, M., "Point-Based Rigid-Body Registration Using an Unscented Kalman Filter," *Transactions on Medical Imaging*, vol. 26, pp. 1708–1728, Dec. 2007.

[63] Mumford, D., "A bayesian rationale for energy functionals," in *Geometry Driven Diffusion in Computer Vision* (Romeny, B., ed.), Kluwer Academic, Dordrecht, pp. 141–153, 1994.

[64] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*. New York, NY: Cambridge University Press, 2003.

[65] Paragios, N. and Deriche, R., "Geodesic active contours and level sets for the detection and tracking of moving objects," in *Transactions on Pattern analysis and Machine Intelligence*, vol. 22, pp. 266–280, 2000.

[66] Paragios, N. and Deriche, R., "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *VCIR*, vol. 13, pp. 249–268, 2002.

[67] Peterfreund, N., "Robust tracking of position and velocity with kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[68] Peterfreundd, N., "The velocity snake: Deformable contour for tracking in spatio-velocity space," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 346–356, 1999.

[69] P.Li, Zhang, T., and Ma, B., "Unscented kalman filter for visual curve tracking," *Image and Vision Computing*, vol. 22, no. 2, pp. 157–164., 2004.

[70] Quan, L. and Lan, Z.-D., "Linear n-point camera pose determination," *IEEE TPAMI*, vol. 21, no. 8, pp. 774–780, 1999.

[71] Rathi, Y., Dambreville, S., and Tannenbaum., A., "Statistical shape analysis using kernel pca," in *Proceedings of SPIE*, vol. 6064, pp. 425–432, 2006.

[72] Rathi, Y., Michailovich, O., Malcolm, J., and Tannenbaum, A., "Seeing the unseen: Segmenting with distributions," in *Proc. Int. Conf. Sig. Imag. Proc.*, 2006.

[73] Rathi, Y., Vaswani, N., Tannenbaum, A., and Yezzi, A., "Tracking deforming objects using particle filtering for geometric active contours," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 8, p. 1470, 2007.

[74] Riklin-Raviv, T., Kiryati, N., and Sochen, N., "Prior-based segmentation by projective registration and level sets," in *ICCV*, pp. 204–211, 2005.

[75] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[76] Rosenhahn, B., Brox, T., and Weickert, J., "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *IJCV*, vol. 73, no. 3, pp. 243–262., 2007.

[77] Rosenhahn, B., Kersting, U., Powel, K., and Seidel, H.-P., "Cloth x-ray: Mocop of people wearing textiles.," in *DAGM*, pp. 495–504, 2006.

[78] Rosenhahn, B., Perwass, C., and Sommer, G., "Pose estimation of free-form contours," *IJCV*, vol. 62, no. 3, pp. 267–289., 2005.

[79] Rother, D. and Sapiro, G., "Seeing 3d Objects in a Single 2d Image," in *ICCV*, 2009.

[80] R.Sandhu, S.Dambreville, and A.Tannenbaum, "Particle filtering for registration of 2d and 3d point sets with stochastic dynamics," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[81] R.Sandhu, S.Dambreville, and A.Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence (Accepted - In Print)*, 2009. Accepted - In Print.

[82] R.Sandhu, S.Dambreville, Yezzi, A., and A.Tannenbaum, "A kernel based framework for non-rigid 2d3d pose estimation and 2d image segmentation," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence (Accepted - In Print), 2010. Accepted - In Print.

[83] R.Sandhu, T.Georgiou, and A.Tannenbaum, "A new distribution metric for image segmentation," in *SPIE Medical Imaging*, 2008.

[84] Salmond, D., Gordon, N., and Smith, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc. F, Radar and signal processing*, vol. 140, pp. 107–113, 1993.

[85] Sandhu, R., Dambreville, S., Yezzi, A., and Tannenbaum, A., "Non-rigid 2d-3d pose estimation and 2d image segmentation," in *CVPR*, p. 2009.

[86] Schmaltz, C., Rosenhahn, B., Brox, T., Cremers, D., Weickert, J., Wietzke, L., and Sommer, G., "Region-based pose tracking," in *Pattern Recognition and Image Analysis*, pp. 56–63, 2007.

[87] Schölkopf, B., Mika, S., and Müller, K., "Nonlinear component analysis as a kernel eigenvalue problem," in *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[88] Sethian, J. A., *Level Set Methods and Fast Marching Methods*. 1999.

[89] Shi, J. and Tomasi, C., "Good features to track," in *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94.*, pp. 593–600, 1994.

[90] Snow, D., Viola, P., and Zabih, R., "Exact Voxel Occupancy with Graph Cuts," in *CVPR*, vol. 1, pp. 345–352, 2000.

[91] Sofka, M., Yang, G., and Stewart, C., "Simultaneous Covariance Driven Correspondence (cdc) and Transformation Estimation in the Expectation Maximization Framework," in *CVPR*, pp. 1–8, 2007.

[92] Stewart, C. V., "Robust Parameter Estimation in Computer Vision," *SIAM Rev.*, vol. 41, no. 3, pp. 513–537, 1999.

[93] Terzopoulos, D. and Szeliski, R., *Tracking with Kalman Snakes.* Active Vision, MIT Press, 1992.

[94] Tsai, A., Yezzi, T., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, E., and Willsky, A., "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE TIP*, vol. 22, no. 2, pp. 137–153, 2003.

[95] Tsin, Y. and Kanade, T., "A Correlation-Based Approach to Robust Point Set Registration," in *ECCV*, pp. 558–569, 2004.

[96] Unal, G., Yezzi, A., Soatto, S., and Slabaugh, G., "A variational approach to problems in calibration of multiple cameras.," *TPAMI*, vol. 29, pp. 1322–1338., 2007.

[97] van der Merwe, R. and Wan, E., "The unscented kalman filter for nonlinear estimation," *Proceedings of IEEE Symposium*, 2000.

[98] Vela, P., Niethammer, M., Pryor, G., Tannenbaum, A., Butts, R., and Washburn, D., "Knowledge-based segmentation for tracking through deep turbulence," *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 469–474, May 2008.

[99] Yang, C., Duraiswami, R., and Davis, L., "Fast multiple object tracking via a hierarchical particle filter," in *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, vol. 1, 2005.

[100] Yezzi, A. and Soatto, S., "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," in *Int. Journal of Computer Vision*, vol. 53, pp. 153–167, 2003.

[101] Yezzi, A. and Soatto, S., "Stereoscopic segmentation.," *IJCV*, vol. 53, no. 3, pp. 31–43., 2003.

[102] Yezzi, A. and Soatto, S., "Structure from motion for scenes without features.," in *Proc. IEEE CVPR*, vol. 1, pp. 171–178, 2003.

[103] Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, 2006.

[104] Zhu, S. C. and Yuille, A., "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 884–900, Sept. 1996.